

# FACE DETECTION USING AWS

## Project Report

**Submitted by**

*POOJHA M-16MIS1039*



**VIT<sup>®</sup>**

**Vellore Institute of Technology**

(Deemed to be University under section 3 of UGC Act, 1956)

CHENNAI

**School of Computer Science and Engineering**

**April 2020**

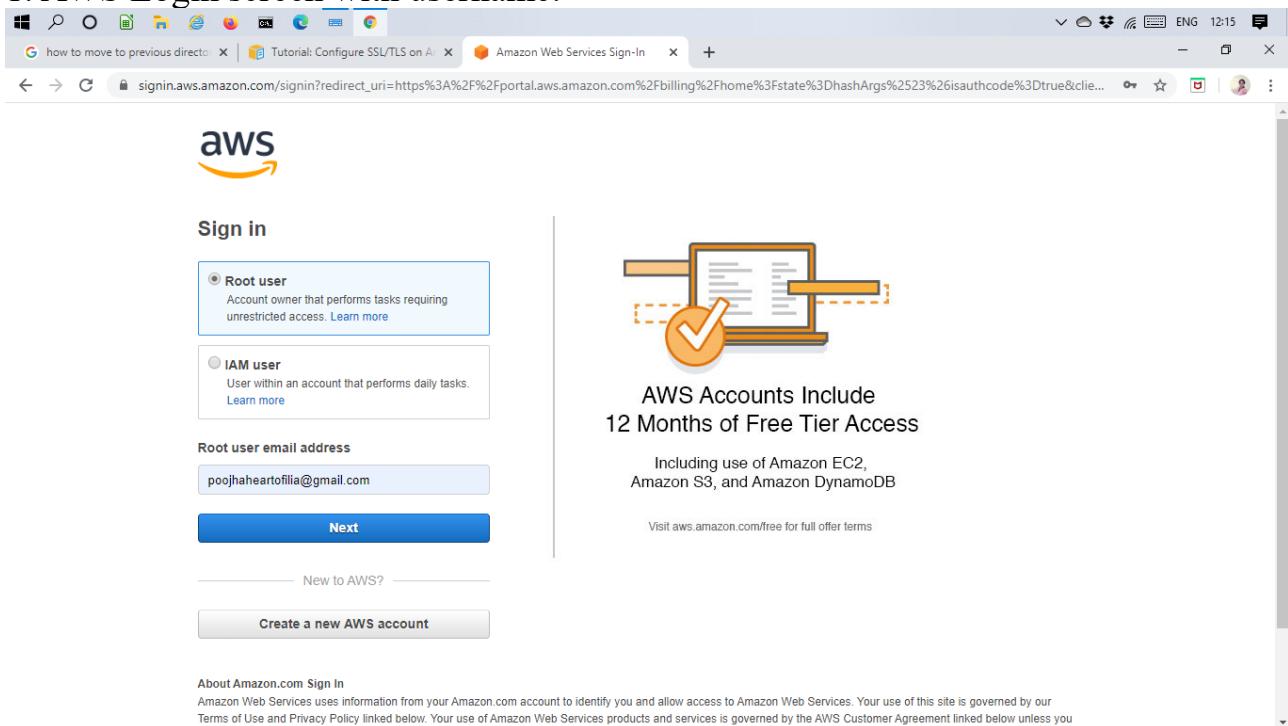
# Table of Content

Topics	Page Number
AWS Login screen with username	1
EC2 Dashboard	1
S3 Dashboard	2
Rekognition Dashboard	2
Choosing an AMI	3
Choosing an Instance Type	3
Adding Storage	4
Configuring Security Group	4
Key Pair Download	5
PuTTYgen conversion from pem to ppk	5
Logged in EC2 black screen	6
Creating a bucket	6
Uploading an Object	7
Enabling Static Website	8
Making the Object Public	9
Checking the S3 link on the browser	9
Face Detect	10
Face Compare	11
Celebrity Recognition	12
Text in Image	12
Installing aws-sdk	13
Installing php	13
index.php file code	14
Upload success screenshot	15
Face Detect success screenshot	16

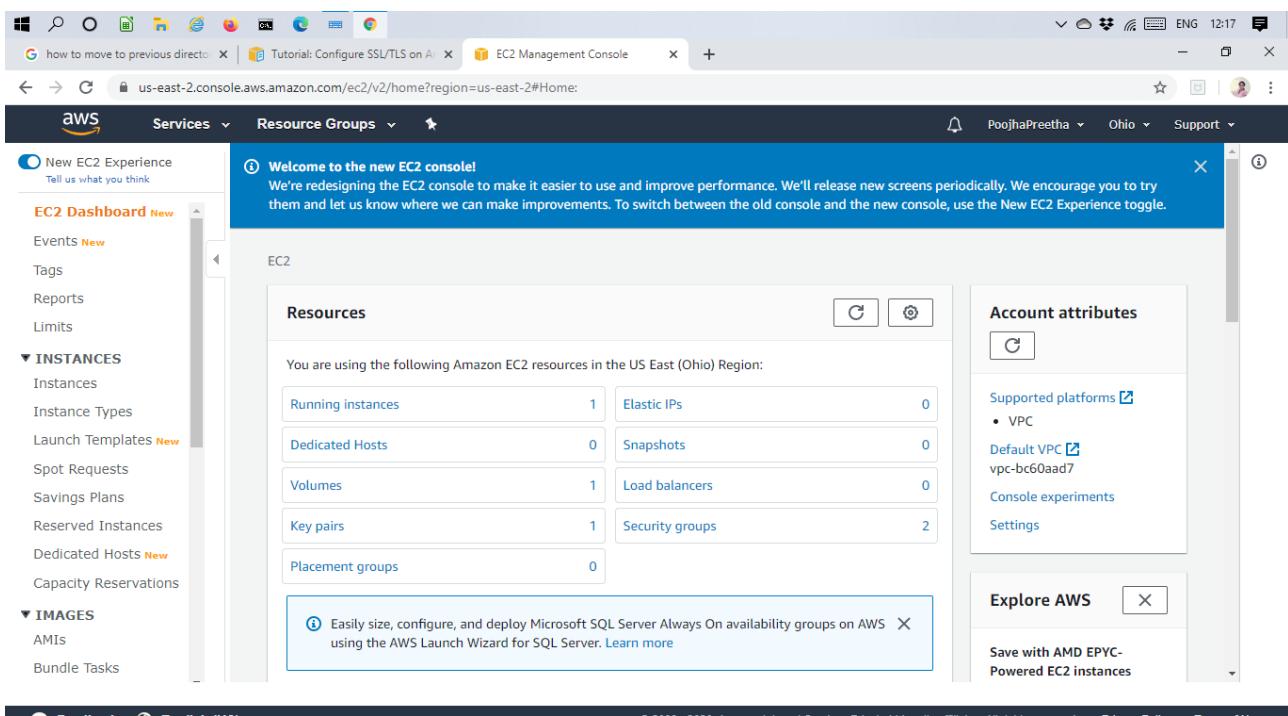
# Screenshots

## Screenshots for Dashboards:

### 1. AWS Login screen with username.



### 2. EC2 Dashboard.



### 3. S3 Dashboard.

The screenshot shows the AWS S3 Management Console. On the left, there's a sidebar for 'Amazon S3' with options like 'Buckets', 'Batch operations', 'Access analyzer for S3', 'Block public access (account settings)', and 'Feature spotlight'. The main area is titled 'Amazon S3' and shows a table for 'Buckets (1)'. The table has columns for Name, Region, Access, and Bucket created. One row is listed: 'aws-webinar-poojha' in US East (Ohio) region, with 'Objects can be public' access and created on '2020-03-27T11:09:46.000Z'. There are buttons for 'Copy ARN', 'Empty', 'Delete', and 'Create bucket'.

### 4. Rekognition Dashboard.

The screenshot shows the AWS Rekognition Console. On the left, there's a sidebar with various services: 'Custom Labels New', 'Use Custom Labels', 'Demos', 'Object and scene detection', 'Image moderation', 'Facial analysis', 'Celebrity recognition', 'Face comparison', 'Text in image', 'Video Demos', 'Video analysis', 'Metrics', and 'Metrics'. The main area has a dark blue background with a network graph. It features a large button labeled 'Try Demo' and another labeled 'Download SDKs'. Below these are three icons: a stack of documents, a circuit board, and puzzle pieces. Text sections include 'Amazon Rekognition Deep learning-based visual analysis service Search, verify, and organize millions of images and videos', 'Easily Integrate Powerful Visual Analysis into Your App You don't need computer vision or deep learning expertise to take advantage of', 'Continuously Learning Amazon Rekognition is designed to use deep learning technology to analyze billions of images and videos daily. It is', and 'Integrated with AWS Services Amazon Rekognition is designed to work seamlessly with other AWS services. Rekognition integrates directly with Amazon'. The bottom navigation bar includes 'Feedback', 'English (US)', 'Privacy Policy', and 'Terms of Use'.

# Screenshots needed for EC2:

## 1. Choosing an AMI

Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Search for an AMI by entering a search term e.g. "Windows"

Quick Start

AMIs
My AMIs
AWS Marketplace
Community AMIs
<input type="checkbox"/> Free tier only ⓘ

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0e01ce4ee18447327 (64-bit x86) / ami-03201f374ab66a26e (64-bit Arm)  
Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras.  
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes  
Select 64-bit (x86)  
64-bit (Arm)

Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type - ami-01b01bbd08f24c7a8  
The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.  
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes  
Select 64-bit (x86)

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

## 2. Choosing an Instance Type

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: All instance types Current generation ShowHide Columns

Family	Type	vCPUs ⓘ	Memory (GiB) ⓘ	Instance Storage (GB) ⓘ	EBS-Optimized Available ⓘ	Network Performance ⓘ	IPv6 Support ⓘ
General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
General purpose	<b>t2.micro</b> Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes

Cancel Previous Review and Launch Next: Configure Instance Details

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

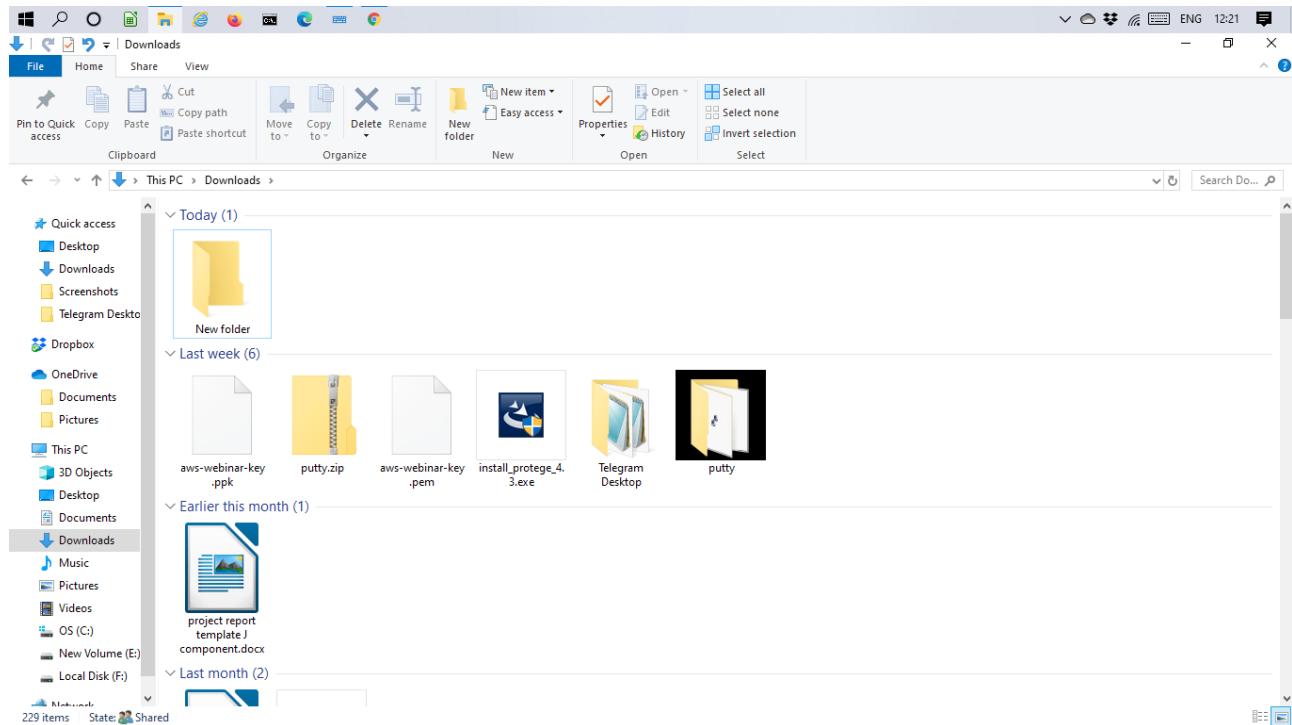
### 3. Adding Storage

The screenshot shows the AWS Launch Instance Wizard at Step 4: Add Storage. The page title is "Step 4: Add Storage". It displays a table for adding storage volumes. A single row is present for the "Root" volume, which is set to "General Purpose SSD (gp2)" with a size of 8 GB. The "Delete on Termination" checkbox is checked, and "Encryption" is set to "Not Encrypted". Below the table, a note states: "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." At the bottom right, there are buttons for "Cancel", "Previous", "Review and Launch", and "Next: Add Tags".

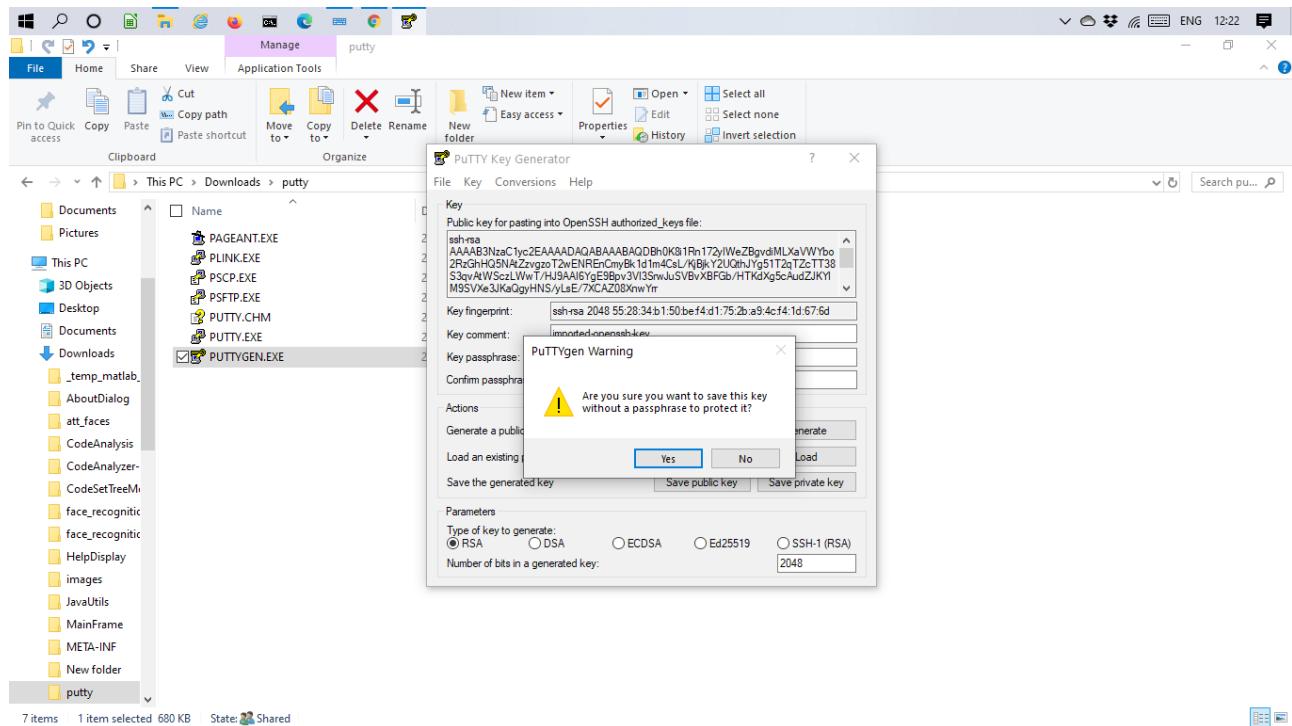
### 4. Configuring Security Group

The screenshot shows the AWS Launch Instance Wizard at Step 6: Configure Security Group. The page title is "Step 6: Configure Security Group". It displays a form for creating a new security group. The "Security group name" is set to "launch-wizard-2" and the "Description" is "launch-wizard-2 created 2020-03-31T12:18:57.550+05:30". A table for defining security rules shows one rule: "SSH" (Type), "TCP" (Protocol), "22" (Port Range), "Custom" (Source), and "0.0.0.0/0" (Description). A warning message in a box states: "⚠️ 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." At the bottom right, there are buttons for "Cancel", "Previous", "Review and Launch", and "Next: Review".

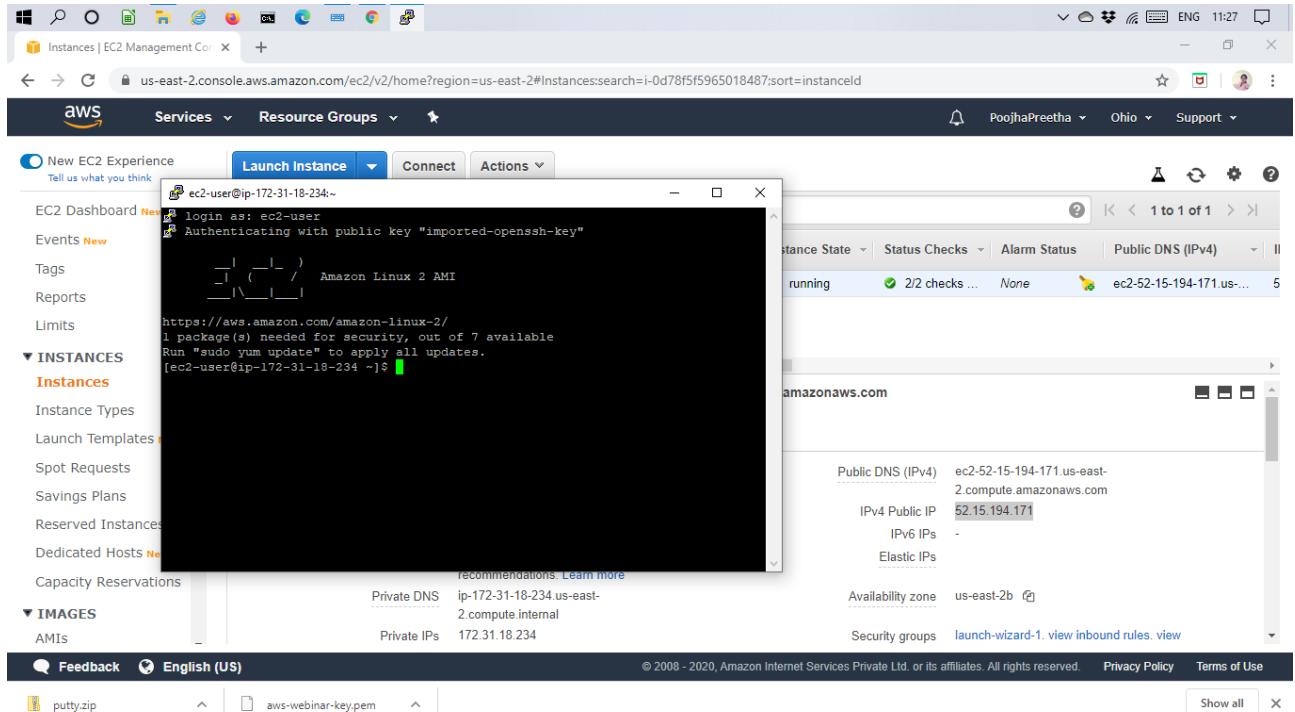
## 5. Key Pair Download



## 6. PuTTYgen conversion from pem to ppk

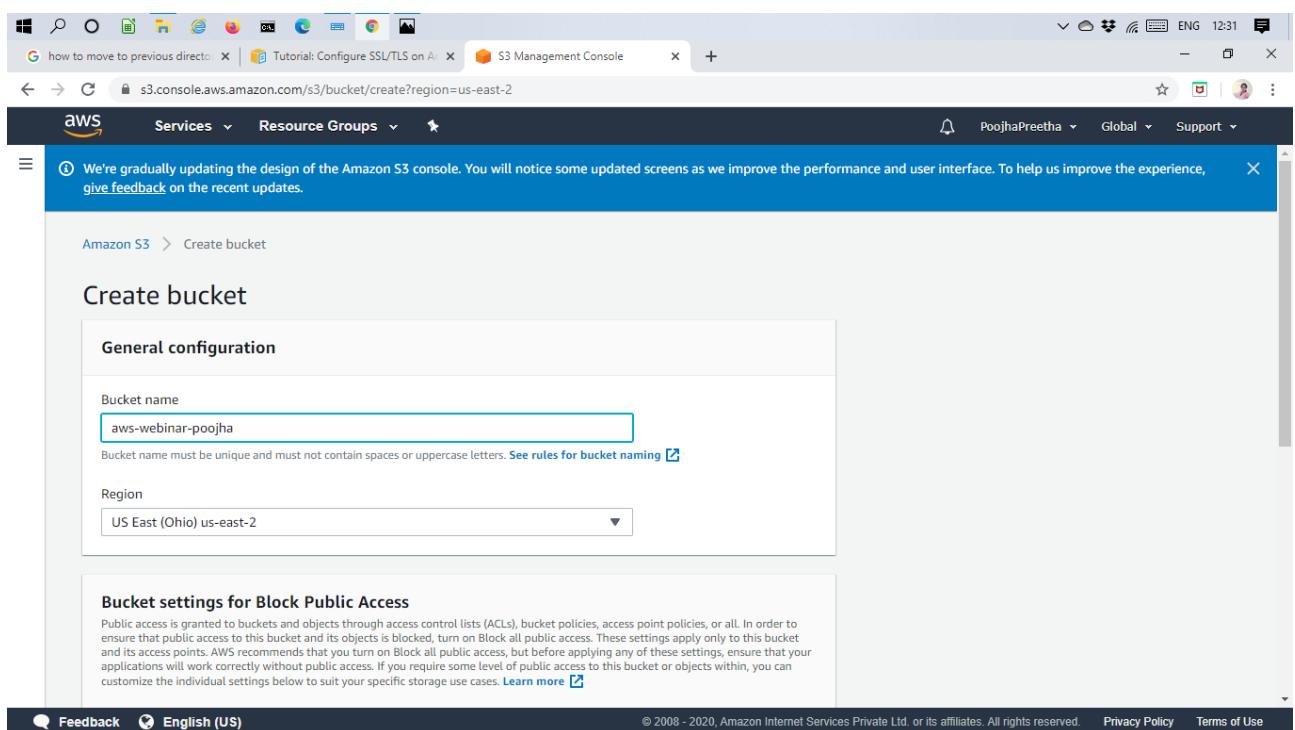


## 7. Logged in EC2 black screen



## Screenshots needed for S3:

### 1. Creating a bucket



The screenshot shows the AWS S3 Management Console interface. On the left, there's a sidebar with options like 'Buckets', 'Batch operations', 'Access analyzer for S3', 'Block public access (account settings)', and 'Feature spotlight'. The main area has a blue banner at the top stating: 'We're gradually updating the design of the Amazon S3 console. You will notice some updated screens as we improve the performance and user interface. To help us improve the experience, give feedback on the recent updates.' Below this is a green success message: 'Successfully created bucket aws-webinar-poojha. To upload files and folders, or to configure additional bucket settings such as Bucket Versioning, tags, and default encryption, choose Go to bucket details.' A 'Go to bucket details' button is shown. The central part of the screen displays the 'Amazon S3' section with a table titled 'Buckets (1)'. The table has columns for Name, Region, Access, and Bucket created. One row is listed: 'aws-webinar-poojha' (Region: US East (Ohio) us-east-2, Access: Not Public, Bucket created: 2020-03-27T11:09:46.000Z). At the bottom, there are links for 'Feedback', 'English (US)', and copyright information.

## 2. Uploading an Object

The screenshot shows the 'aws-webinar-poojha' bucket details page in the AWS S3 Management Console. The top navigation bar includes 'Services', 'Resource Groups', and the user 'PoojhaPreetha'. The left sidebar shows the bucket name. The main content area has tabs for 'Overview', 'Properties', 'Permissions', 'Management', and 'Access points', with 'Management' selected. Below is a search bar and a toolbar with 'Upload', 'Create folder', 'Download', and 'Actions'. The main table lists three objects: 'index.html' (Last modified: Mar 27, 2020 4:51:42 PM, Size: 127.0 B, Storage class: Standard), 's.jpg' (Last modified: Mar 29, 2020 3:53:35 PM, Size: 210.5 KB, Storage class: Standard), and 'sample.jpg' (Last modified: Mar 30, 2020 3:14:31 PM, Size: 210.5 KB, Storage class: Standard). The table has columns for Name, Last modified, Size, and Storage class. At the bottom, there are links for 'Feedback', 'English (US)', and copyright information.

### 3. Enabling Static Website

The screenshot shows the AWS S3 Management Console with the 'Static website hosting' configuration page open. The 'Endpoint' is listed as <http://aws-webinar-poojha.s3-website.us-east-2.amazonaws.com>. Under 'Index document', 'index.html' is selected. Under 'Error document', 'error.html' is selected. There are sections for 'Redirection rules (optional)' and 'Bucket hosting'. A modal window titled 'Object-level logging' is displayed, stating: 'Record object-level API activity using the CloudTrail data events feature (additional cost)'. It has a 'Learn more' link and a radio button set to 'Disabled'. At the bottom are 'Cancel' and 'Save' buttons.

The screenshot shows the AWS S3 Management Console with the bucket configuration page for 'aws-webinar-poojha'. The 'Access points' tab is selected. It displays several features: 'Versioning' (disabled), 'Server access logging' (disabled), 'Static website hosting' (selected, with a checked checkbox and 'Bucket hosting' checked), 'Object-level logging' (disabled), and 'Default encryption' (disabled). The bottom navigation bar includes 'Feedback', 'English (US)', 'Privacy Policy', and 'Terms of Use'.

## 4. Making the Object Public

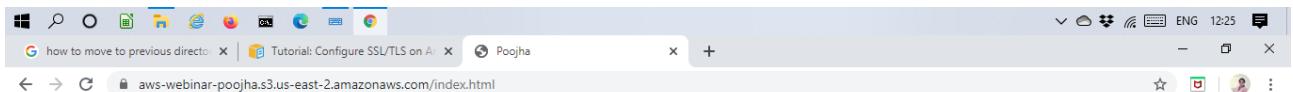
The screenshot shows the AWS S3 Management Console with the 'Block public access' tab selected. The 'Block all public access' setting is currently off. There are four sub-options under it, each also set to off:

- Block public access to buckets and objects granted through *new* access control lists (ACLs)
- Block public access to buckets and objects granted through *any* access control lists (ACLs)
- Block public access to buckets and objects granted through *new* public bucket or access point policies
- Block public and cross-account access to buckets and objects through *any* public bucket or access point policies

At the bottom right of the main panel, there is an 'Edit' button.

## 5. Checking the S3 link on the browser

The screenshot shows the AWS S3 Management Console displaying the details of an object named 'index.html'. The object was last modified on Mar 27, 2020 at 4:51:42 PM GMT+0530. It has a size of 127.0 B and is stored in the 'Standard' storage class. The object URL is provided as a link: <https://aws-webinar-poojha.s3.us-east-2.amazonaws.com/index.html>.



Hello

Hello

## Screenshots for Rekognition:

### 1. Face Detect

Get a complete analysis of facial attributes, including confidence scores.

Done with the demo? [Learn more](#)

Results

Attribute	Confidence Score
looks like a face	99.9 %
appears to be female	99.9 %
age range	17 - 29 years old
smiling	91.7 %
appears to be happy	99.5 %
wearing glasses	99.8 %
wearing sunglasses	92.2 %

Choose a sample image

Use your own image  
Image must be .jpeg or .png format and no larger than 5MB. Your image isn't stored.

Upload or drag and drop

Use image URL Go

Feedback English (US)

© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Screenshot of the AWS Rekognition Console showing Face Detection results for a photo of a woman driving a yellow car.

**Face Detection Results:**

looks like a face	99.9 %
appears to be female	99.9 %
age range	17 - 29 years old
smiling	91.7 %
appears to be happy	99.5 %
wearing glasses	99.8 %
wearing sunglasses	92.2 %
eyes are open	99.9 %
mouth is open	99.5 %
does not have a mustache	99.9 %
does not have a beard	99.8 %

**Left Sidebar:**

- Amazon Rekognition
- Custom Labels New
- Use Custom Labels
- Demos
- Object and scene detection
- Image moderation
- Facial analysis** (selected)
- Celebrity recognition
- Face comparison
- Text in image
- Video Demos
- Video analysis
- Metrics
- Metrics

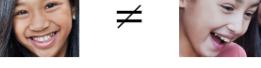
**Bottom Navigation:**

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

## 2. Face Compare

Screenshot of the AWS Rekognition Console showing Face Comparison results.

**Face Comparison Results:**

Comparison	Similarity
 vs 	99.9 %
 vs 	99.9 %
 vs 	99.9 %

**Left Sidebar:**

- Amazon Rekognition
- Custom Labels New
- Use Custom Labels
- Demos
- Object and scene detection
- Image moderation
- Facial analysis
- Celebrity recognition
- Face comparison** (selected)
- Text in image
- Video Demos
- Video analysis
- Metrics
- Metrics

**Bottom Navigation:**

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

### 3. Celebrity Recognition

The screenshot shows the Amazon Rekognition console under the 'Celebrity recognition' demo. On the left sidebar, 'Celebrity recognition' is selected. In the center, a portrait of Jeff Bezos is displayed with a blue bounding box around his face. Below it are options to 'Choose a sample image' or 'Upload your own image'. To the right, the results show a small thumbnail of Jeff Bezos with the name 'Jeff Bezos' and a 'Match confidence' of 100%. There are also sections for 'Request' and 'Response'.

### 4. Text in Image

The screenshot shows the Amazon Rekognition console under the 'Text in image' demo. On the left sidebar, 'Text in image' is selected. In the center, a photo of a coffee mug with text overlays ('IT'S MONDAY but keep Smiling') is shown. Below it are options to 'Choose a sample image' or 'Upload your own image'. To the right, the results show detected text: 'IT'S', 'MONDAY', 'but', 'keep', and 'Smiling'. The results are limited to 'US English only'.

## Screenshots for EC2 & S3:

### 1. Installing aws-sdk

```
ec2-user@ip-172-31-18-234:/var/www/html/face$ PHP Warning: proc_open(): fork failed - Cannot allocate memory in phar:///home/ec2-user/composer.phar/vendor/symfony/console/Application.php on line 952
Warning: proc open(): fork failed - Cannot allocate memory in phar:///home/ec2-user/composer.phar/vendor/symfony/console/Application.php on line 952
[ErrorException]
proc_open(): fork failed - Cannot allocate memory

require [--dev] [--prefer-source] [--prefer-dist] [--fixed] [--no-progress] [--no-suggest] [--no-update] [--no-scripts] [--update-no-dev] [--update-with-dependencies] [--update-with-all-dependencies] [--ignore-platform-reqs] [--prefer-stable] [--prefer-lowest] [--sort-packages] [-o|--optimize-autoloader] [-a|--classmap-authoritative] [--apcu-autoloader] [--] {packages}...
[ec2-user@ip-172-31-18-234 face]$ sudo /bin/dd if=/dev/zeroof=/var/swapl.1 bs=1M count=1024
/bin/dd: failed to open '/dev/zeroof=/var/swapl.1': No such file or directory
[ec2-user@ip-172-31-18-234 face]$ sudo /bin/dd if=/dev/zero of=/var/swapl.1 bs=1M count=1024
1024+0 records in
1024+0 records out
1073741824 bytes (1.1 GB) copied, 13.4352 s, 79.9 MB/s
[ec2-user@ip-172-31-18-234 face]$ sudo /sbin/mkswap /var/swapl.1
mkswap: /var/swapl.1: insecure permissions 0644, 0600 suggested.
Setting up swapspace version 1, size = 1024 MiB (1073737728 bytes)
no label, UUID=be519d61-f86d-4ee9-8dc8-25bedf260b9c
[ec2-user@ip-172-31-18-234 face]$ sudo /sbin/swapon /var/swapl.1
swapon: /var/swapl.1: insecure permissions 0644, 0600 suggested.
[ec2-user@ip-172-31-18-234 face]$ sudo php -d memory_limit=-1 ~/composer.phar require aws/aws-sdk-php
Using version 2.8 for aws/aws-sdk-php
./composer.json has been created
Loading composer repositories with package information
Updating dependencies (including require-dev)
Package operations: 3 installs, 0 updates, 0 removals
- Installing symfony/event-dispatcher (v2.8.52): Loading from cache
- Installing guzzle/guzzle (v3.9.3): Downloading (100%)
- Installing aws/aws-sdk-php (2.8.31): Downloading (100%)
symfony/event-dispatcher suggests installing symfony/dependency-injection
symfony/event-dispatcher suggests installing symfony/http-kernel
guzzle/guzzle suggests installing guzzlehttp/guzzle (Guzzle 5 has moved to a new package name. The package you have installed, Guzzle 3, is deprecated.)
aws/aws-sdk-php suggests installing doctrine/cache (Adds support for caching of credentials and responses)
aws/aws-sdk-php suggests installing ext-apc (Allows service description opcode caching, request and response caching, and credentials caching)
aws/aws-sdk-php suggests installing monolog/monolog (Adds support for logging HTTP requests and responses)
aws/aws-sdk-php suggests installing symfony/yaml (Eases the ability to write manifests for creating jobs in AWS Import/Export)
Package guzzle/guzzle is abandoned, you should avoid using it. Use guzzlehttp/guzzle instead.
Writing lock file
Generating autoload files
[ec2-user@ip-172-31-18-234 face]$
```

### 2. Installing php

```
ec2-user@ip-172-31-18-234:~
1 package(s) needed for security, out of 7 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-18-234 ~]$ sudo yum install httpd
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
| 2.4 kB   00:00
Package httpd-2.4.41-1.amzn2.0.1.x86_64 already installed and latest version
Nothing to do
[ec2-user@ip-172-31-18-234 ~]$ sudo yum install php
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
--> Package php.x86_64 0:5.4.16-46.amzn2.0.2 will be installed
--> Processing Dependency: php-cli(x86-64) = 5.4.16-46.amzn2.0.2 for package: php
--> Processing Dependency: php-common(x86-64) = 5.4.16-46.amzn2.0.2 for package: php
--> Processing Dependency: libzip.so.2() (64bit) for package: php-common-5.4.16-4
6.amzn2.0.2.x86_64
--> Running transaction check
--> Package php-cli.x86_64 0:5.4.16-46.amzn2.0.2 will be installed
--> Package php-common.x86_64 0:5.4.16-46.amzn2.0.2 will be installed
--> Processing Dependency: libzip.so.2() (64bit) for package: php-common-5.4.16-4
6.amzn2.0.2.x86_64
--> Running transaction check
--> Package libzip010-compat.x86_64 0:0.10.1-9.amzn2.0.5 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
 Package           Arch      Version            Repository      Size
=====
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]:
```

```

ec2-user@ip-172-31-18-234:~$ sudo apt update
[...]
ec2-user@ip-172-31-18-234:~$ sudo apt install php
[...]
ec2-user@ip-172-31-18-234:~$ ls -l /var/www/html
total 0
ec2-user@ip-172-31-18-234:~$ curl http://172.31.18.234/index.php
[...]

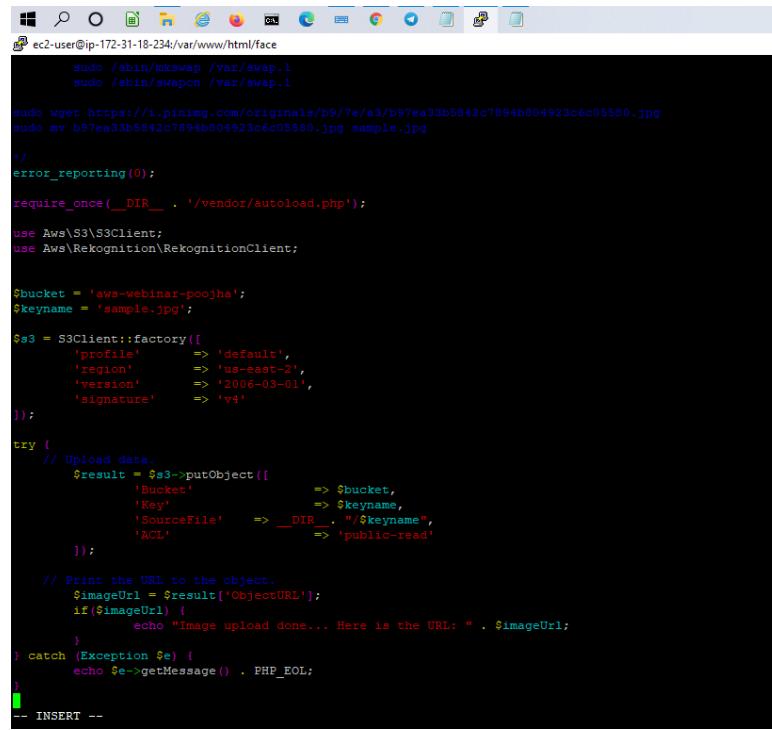
```

### 3. index.php file code

```

index(2).php
C:\Users\ROOJHA\Downloads\Telegram Desktop> index(2).php
1 <?php
2 // error_reporting(0);
3 require_once(__DIR__ . '/vendor/autoload.php');
4 use Aws\S3\S3Client;
5 use Aws\Rekognition\RekognitionClient;
6 $bucket = 'aws-webinar-pooja';
7 $keyname = 'sample.jpg';
8
9 $s3 = new S3Client([
10     'region' => 'us-east-2',
11     'version' => '2006-03-01',
12     'signature' => 'v4'
13 ]);
14
15 try {
16     // Upload data.
17     $result = $s3->putObject([
18         'Bucket' => $bucket,
19         'Key' => $keyname,
20         'SourceFile' => __DIR__ . "/$keyname",
21         'ACL' => 'public-read-write'
22     ]);
23
24     // Print the URL to the object.
25     $imageUrl = $result['ObjectURL'];
26     if($imageUrl) {
27         echo "Image upload done... Here is the URL: " . $imageUrl;
28
29     $rekognition = new RekognitionClient([
30         'region' => 'us-east-2',
31         'version' => 'latest',
32     ]);
33
34     $result = $rekognition->detectFaces([
35         'Attributes' => ['DEFAULT'],
36         'Image' => [
37             'S3Object' => [
38                 'Bucket' => $bucket,
39                 'Name' => $keyname,
40                 'Key' => $keyname,
41             ],
42         ],
43     ]);
44
45     echo "Totally there are " . count($result["FaceDetails"]) . " faces";
46 }
47 } catch (Exception $e) {
48     echo $e->getMessage() . PHP_EOL;
49 }

```



```
ec2-user@ip-172-31-18-234:~/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/b97ea33b5842c7894b804503c6c05580.jpg
sudo mv b97ea33b5842c7894b804503c6c05580.jpg sample.jpg

/*
error_reporting(0);

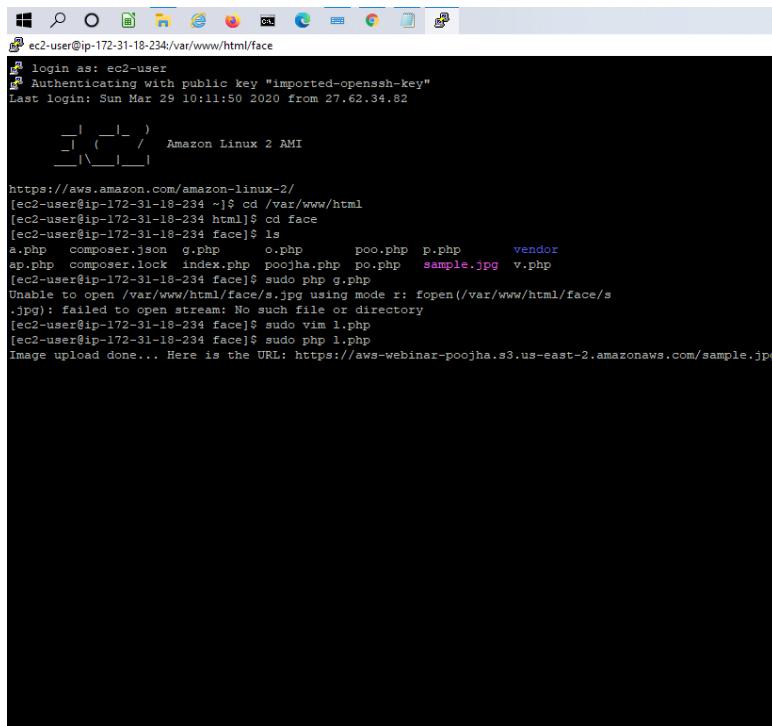
require_once(__DIR__ . '/vendor/autoload.php');

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

$bucket = 'aws-webinar-poojha';
$keyname = 'sample.jpg';

$s3 = S3Client::factory([
    'profile'      => 'default',
    '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'
    ]);
    // Print the URL to the object.
    $imageUrl = $result['ObjectURL'];
    if($imageUrl) {
        echo "Image upload done... Here is the URL: " . $imageUrl;
    }
} catch (Exception $e) {
    echo $e->getMessage() . PHP_EOL;
}
-- INSERT --
```

#### 4. Upload success screenshot



```
ec2-user@ip-172-31-18-234:~/var/www/html/face
[ec2-user@ip-172-31-18-234 ~]$ login as: ec2-user
[ec2-user@ip-172-31-18-234 ~]$ Authenticating with public key "imported-openssh-key"
Last login: Sun Mar 29 10:11:50 2020 from 27.62.34.82
[ec2-user@ip-172-31-18-234 ~]$ 
[ec2-user@ip-172-31-18-234 ~]$ ls
[ec2-user@ip-172-31-18-234 ~]$ cd /var/www/html
[ec2-user@ip-172-31-18-234 html]$ cd face
[ec2-user@ip-172-31-18-234 face]$ ls
a.php  composer.json  g.php  o.php  poo.php  p.php  vendor
ap.php  composer.lock  index.php  poojha.php  po.php  sample.jpg  v.php
[ec2-user@ip-172-31-18-234 face]$ sudo php g.php
Unable to open /var/www/html/face/s.jpg using mode r: fopen(/var/www/html/face/s.jpg): failed to open stream: No such file or directory
[ec2-user@ip-172-31-18-234 face]$ sudo vim l.php
[ec2-user@ip-172-31-18-234 face]$ sudo php l.php
Image upload done... Here is the URL: https://aws-webinar-poojha.s3.us-east-2.amazonaws.com/sample.jpg
```

# Screenshots for EC2 & Rekognition:

## 1. Face Detect success screenshot

```
ec2-user@ip-172-31-18-234:/var/www/html/face$ login as: ec2-user
Authenticating with public key "imported-openssh-key"
Last login: Mon Mar 30 07:22:42 2020 from 27.62.40.200
[ec2-user@ip-172-31-18-234 ~]$ cd /var/www/html
[ec2-user@ip-172-31-18-234 html]$ cd face
[ec2-user@ip-172-31-18-234 face]$ sudo vim l.php
[ec2-user@ip-172-31-18-234 face]$ sudo php l.php
PHP Parse error: syntax error, unexpected end of file in /var/www/html/face/l.php on line 78
[ec2-user@ip-172-31-18-234 face]$ sudo vim face.php
[ec2-user@ip-172-31-18-234 face]$ sudo php face.php
Image upload done... Here is the URL: https://aws-webinar-poojha.s3.us-east-2.amazonaws.com/sample.jpgTotally there are 9 faces[ec2-user@ip-172-31-18-234 face]$
```

