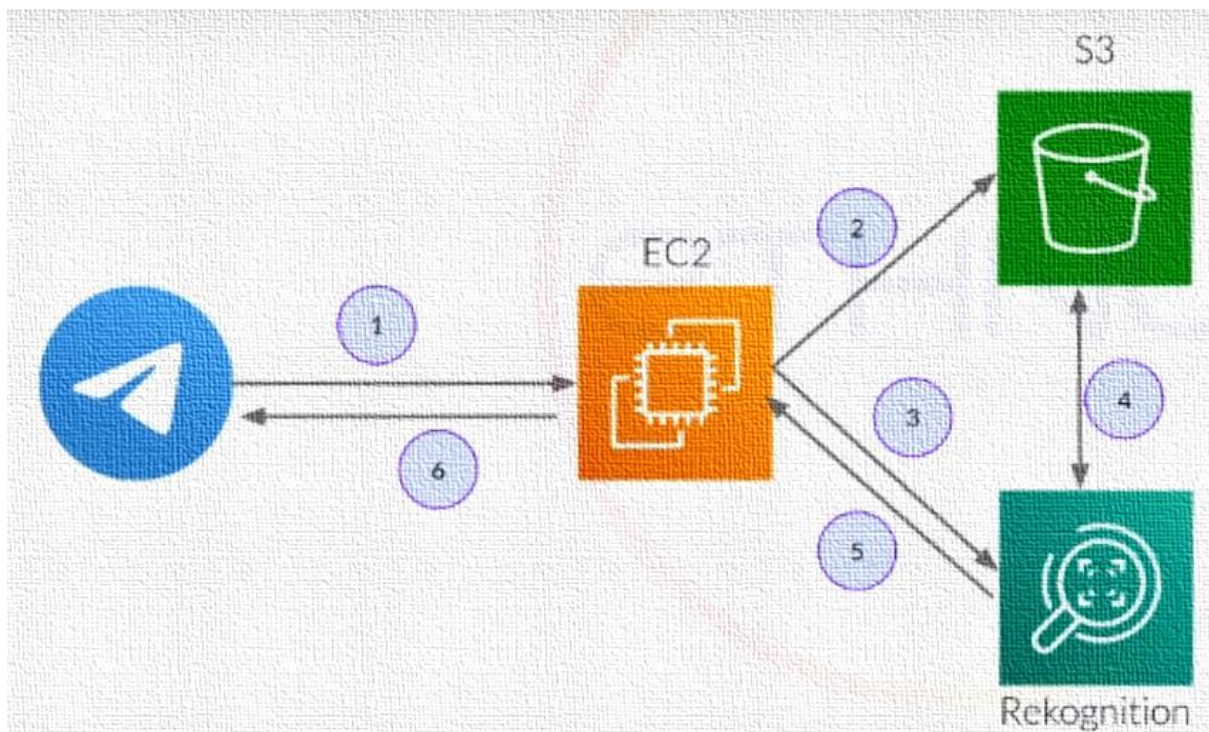


Face Detection using Amazon Web Services

Services Used to build the application:

1. EC2 – Elastic Compute Cloud
2. S3 – Simple Storage Service
3. Rekognition
4. Telegram Bot

Application Architecture:



Dashboard Screenshots for all the used Services:

a) AWS Login screen with username:



Sign in

☒ **Root user**

Account owner that performs tasks requiring unrestricted access. [Learn more](#)

☐ **IAM user**

User within an account that performs daily tasks. [Learn more](#)

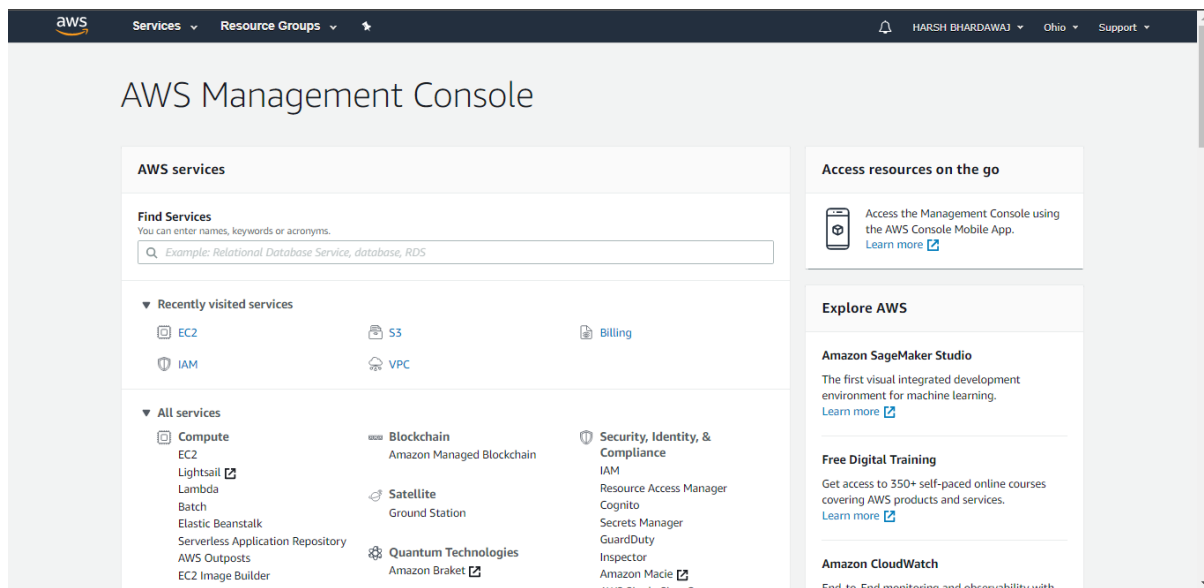
Root user email address

harshbhardawaj@outlook.com

Next

New to AWS?

Create a new AWS account



b) EC2 Dashboard:

The screenshot shows the AWS Management Console for the EC2 service. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', and a user profile 'HARSH BHARDWAJ' in the 'Ohio' region. The left sidebar contains a navigation menu with categories like 'New EC2 Experience', 'Events', 'Tags', 'Reports', 'Limits', 'INSTANCES', 'IMAGES', and 'ELASTIC BLOCK STORE'. The main content area is titled 'EC2' and displays a 'Resources' section with a table of EC2 resources in the 'US East (Ohio) Region':

Resources	
Running instances	1
Elastic IPs	0
Dedicated Hosts	0
Snapshots	0
Volumes	1
Load balancers	0
Key pairs	1
Security groups	3
Placement groups	0

Below the table, there is a 'Launch instance' button and a 'Service health' section. On the right, there are panels for 'Account attributes' (showing supported platforms like VPC) and 'Explore AWS' (with links to optimize EC2 cost and performance with Spot Instances).

c) S3 Dashboard:

The screenshot shows the AWS Management Console for the S3 service. The top navigation bar is similar to the EC2 dashboard, with the user profile 'HARSH BHARDWAJ' in the 'Global' region. The left sidebar shows the 'Amazon S3' section with options like 'Buckets', 'Batch operations', and 'Access analyzer for S3'. The main content area is titled 'Amazon S3' and displays a 'Buckets (1)' section. It includes a search bar and a table of buckets:

Name	Region	Access	Bucket created
aws--webinar	US East (Ohio) us-east-2	Objects can be public	2020-03-28T04:56:32.000Z

At the top of the bucket list, there are buttons for 'Copy ARN', 'Empty', 'Delete', and 'Create bucket'. The bottom of the console shows a footer with 'Feedback', 'English (US)', and copyright information.

d) Rekognition Dashboard:

Amazon Rekognition

Deep learning-based visual analysis service
Search, verify, and organize millions of images and videos

[Try Demo](#)
[Download SDKs](#)

Easily Integrate Powerful Visual Analysis into Your App
You don't need computer vision or deep learning expertise to take advantage of Rekognition's high quality image and video analysis for your web, mobile, enterprise or device applications. Amazon Rekognition removes the complexity of building

Continuously Learning
Amazon Rekognition is designed to use deep learning technology to analyze billions of images and videos daily. It is continuously learning as we add support for new capabilities and learn from more and more data.

Integrated with AWS Services
Amazon Rekognition is designed to work seamlessly with other AWS services. Rekognition integrates directly with Amazon S3 and AWS Lambda so you can build scalable, affordable, and reliable visual analysis applications. You can start analyzing images and videos stored in Amazon S3

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

EC2 Configuration Screenshots:

a) 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

- My AMIs
- AWS Marketplace
- Community AMIs
- ☐ 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

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

Red Hat Enterprise Linux 8 (HVM), SSD Volume Type - ami-0520e698dd500b1d1 (64-bit x86) / ami-0099847d600887c9f (64-bit Arm)

Red Hat Enterprise Linux version 8 (HVM), EBS General Purpose (SSD) Volume Type

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

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

b) 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 Show/Hide Columns

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs ⓘ	Memory (GiB)	Instance Storage (GB) ⓘ	EBS-Optimized Available ⓘ	Network Performance ⓘ	IPv6 Support ⓘ
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.xlarge	4	16	EBS only	-	Moderate	Yes
<input type="checkbox"/>	General purpose	t2.2xlarge	8	32	EBS only	-	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

c) 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](#)

d) Configure 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](#)

e) Review of Instance Launch:

Step 7: Review Instance Launch

AMI Details

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0e01ce4ee18447327

Free tier eligible

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

Instance Type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

Security Groups

Security group name: launch-wizard-3

Description: launch-wizard-3 created 2020-03-31T15:39:10.476+05:30

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	0.0.0.0/0	

Instance Details

Cancel Previous Launch

f) Selecting/Downloading a key Pair:

Step 7: Review Instance Launch

Root Device Type: ebs Virtualization type: hvm

Instance Type

Instance Type	ECUs	vCPUs
t2.micro	Variable	1

Security Groups

Security group name: launch-wizard-3

Description: launch-wizard-3 created 2020-03-31T15:39:10.476+05:30

Type	Protocol
SSH	TCP

Instance Details

Storage

Tags

Cancel Previous Launch

Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Choose an existing key pair

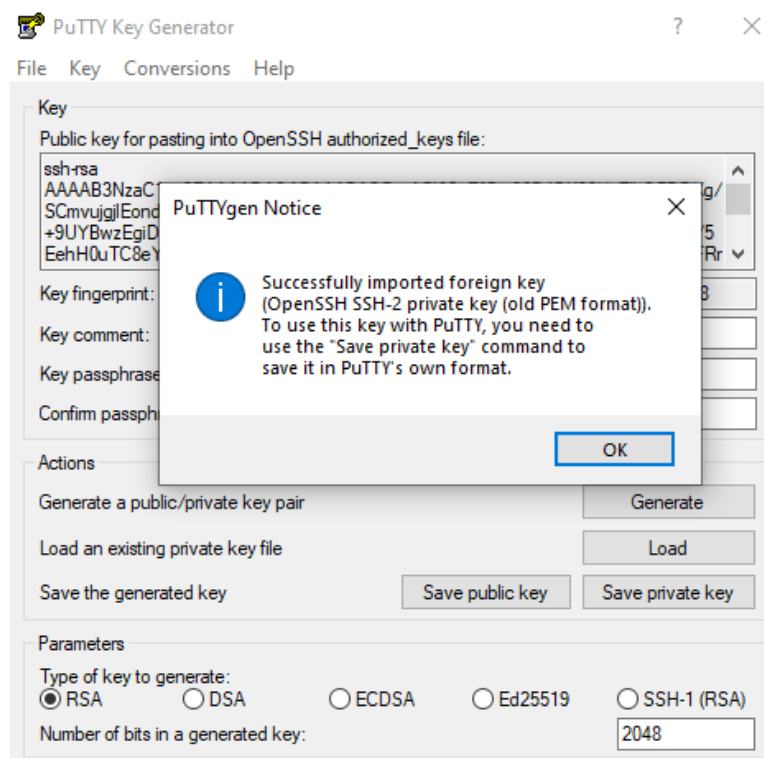
Select a key pair

aws-webinar-key

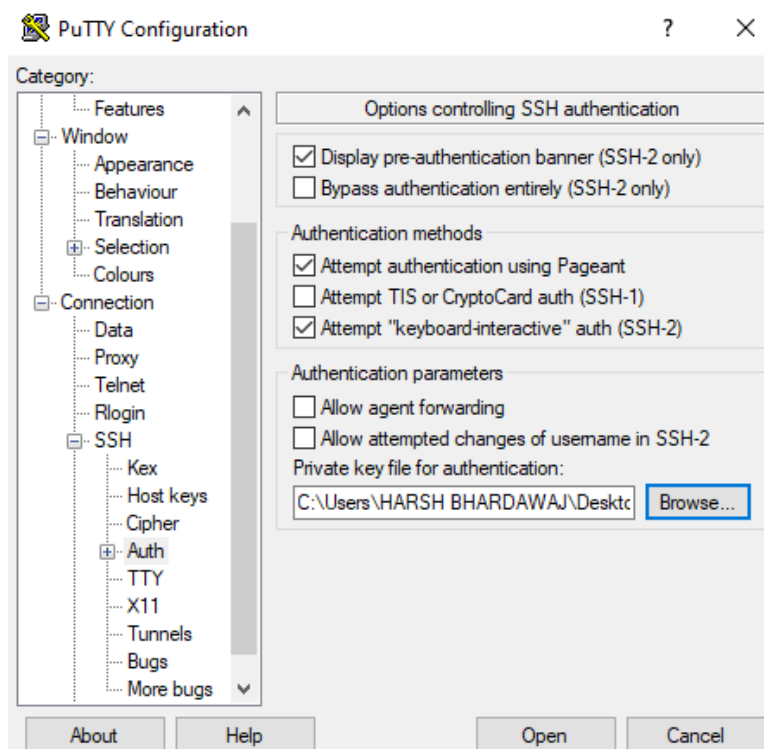
☒ I acknowledge that I have access to the selected private key file (aws-webinar-key.pem), and that without this file, I won't be able to log into my instance.

Cancel Launch Instances

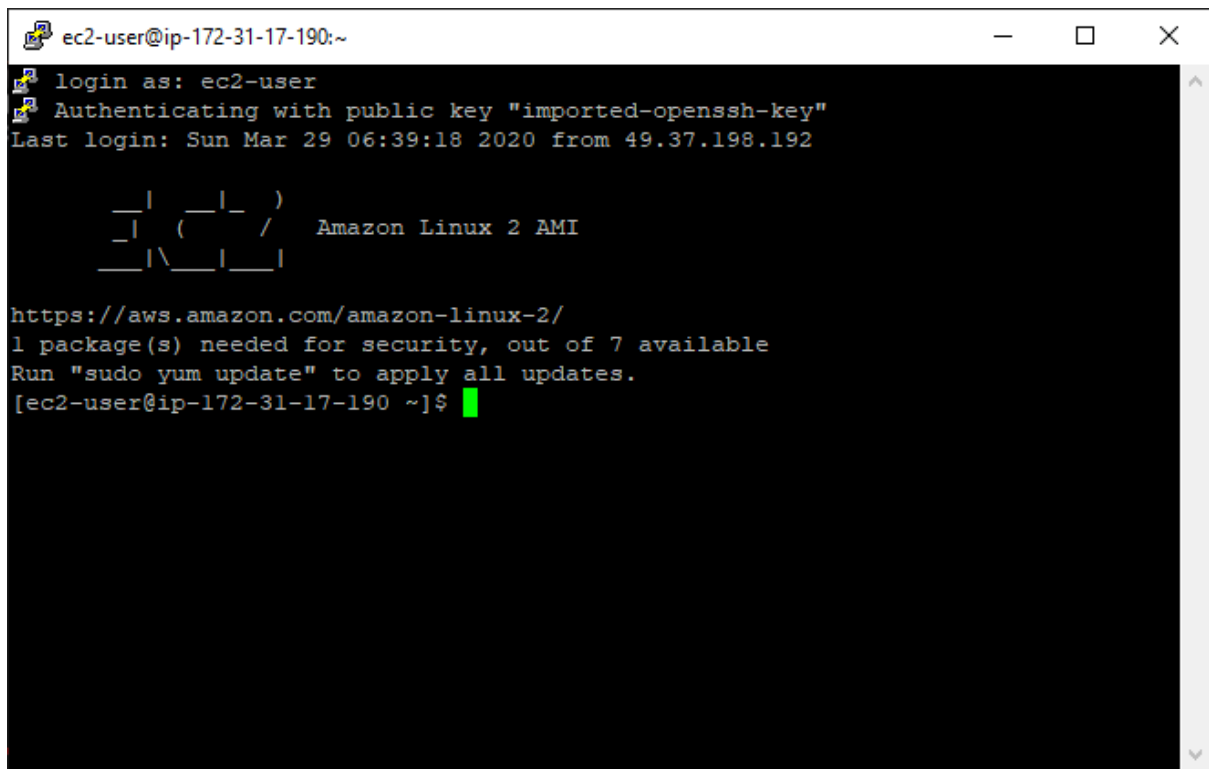
g) Puttygen Conversion of key from pem to ppk format:



h) Remotely Accessing EC2 using private key:



i) Logged in EC2 Screen:



```
ec2-user@ip-172-31-17-190:~  
login as: ec2-user  
Authenticating with public key "imported-openssh-key"  
Last login: Sun Mar 29 06:39:18 2020 from 49.37.198.192  
  
  _ | _ | _ )  
  _ | ( _ /  Amazon Linux 2 AMI  
  _ | \ _ | _ |  
  
https://aws.amazon.com/amazon-linux-2/  
1 package(s) needed for security, out of 7 available  
Run "sudo yum update" to apply all updates.  
[ec2-user@ip-172-31-17-190 ~]$
```

S3 Configuration Screenshots:

a) Creating a Bucket:

Amazon S3 Create bucket

General configuration

Bucket name
aws--webinar
Bucket name must be unique and must not contain spaces or uppercase letters. [See rules for bucket naming](#)

Region
US East (Ohio) us-east-2

Bucket settings for Block Public Access

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

☒ **Block all public access**
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

☒ **Block public access to buckets and objects granted through new access control lists (ACLs)**
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.

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

b) Successful Bucket Creation:

Amazon S3 Successfully created bucket aws--webinar
To upload files and folders, or to configure additional bucket settings such as Bucket Versioning, tags, and default encryption, choose [Go to bucket details](#).

Amazon S3

Buckets (1) [Copy ARN](#) [Empty](#) [Delete](#) [Create bucket](#)

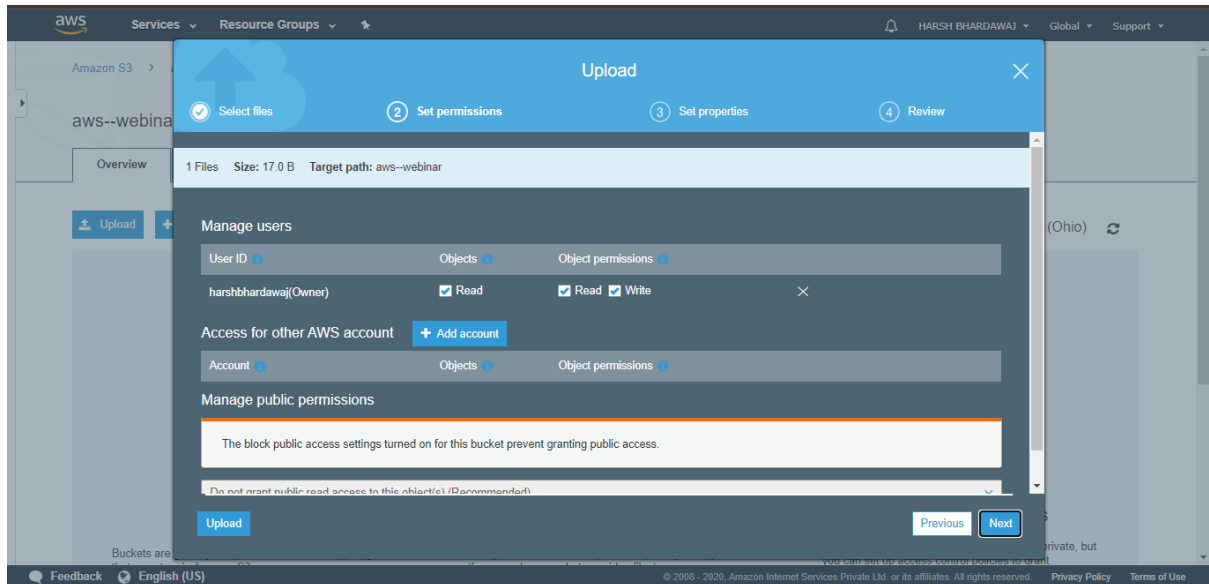
Find bucket by name

	Name	Region	Access	Bucket created
<input type="radio"/>	aws--webinar	US East (Ohio) us-east-2	Not Public	2020-03-28T04:56:32.000Z

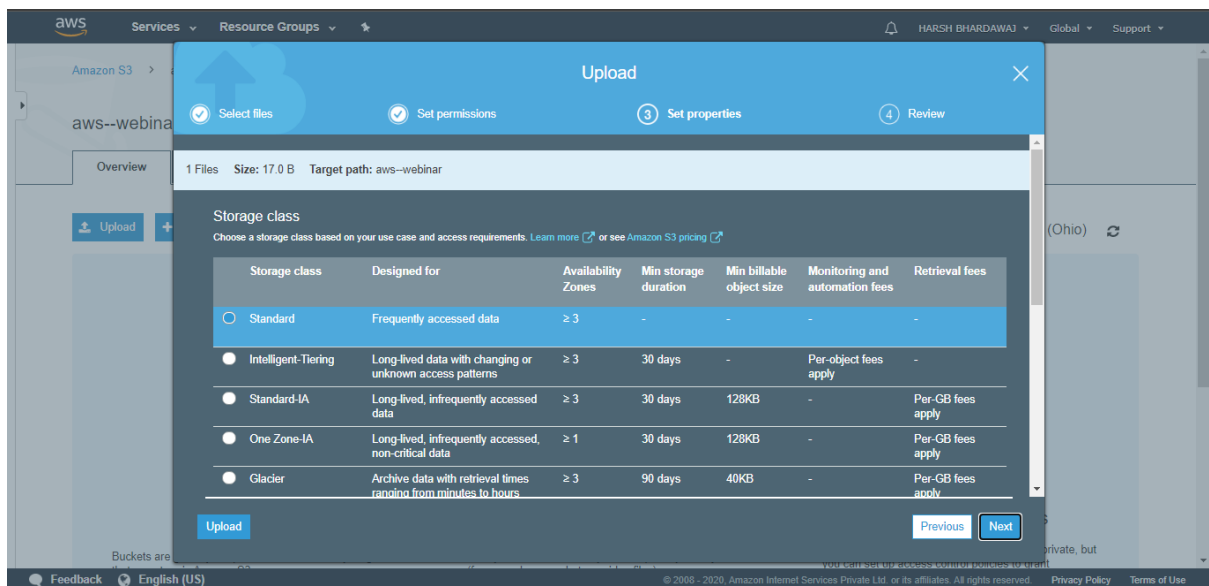
Feedback English (US) © 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

c) Uploading an object:

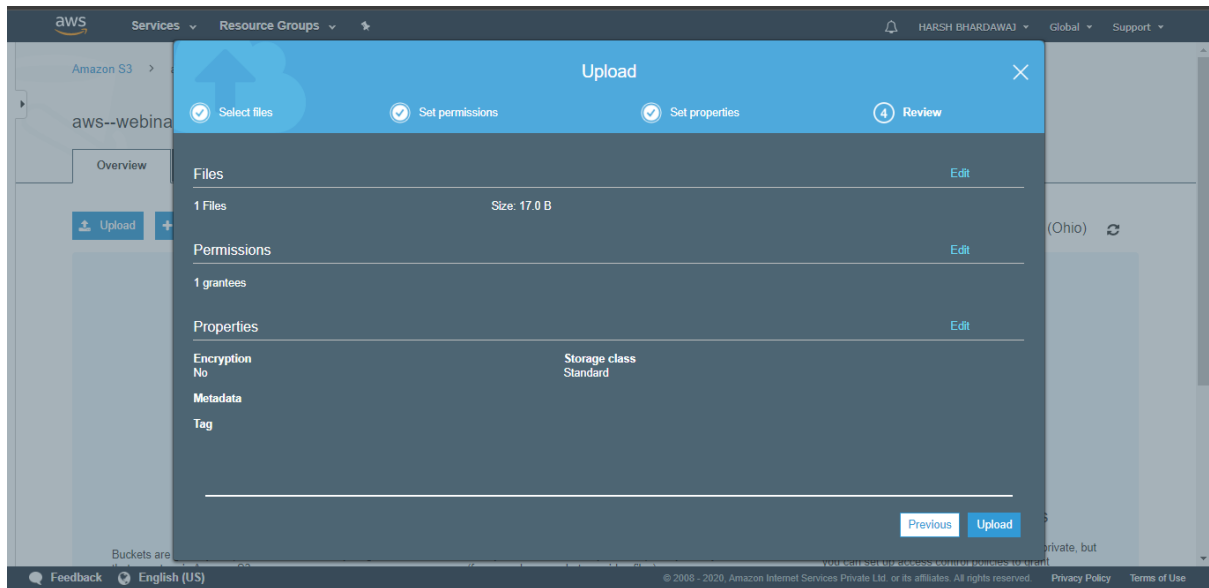
Step i: Selecting a file:



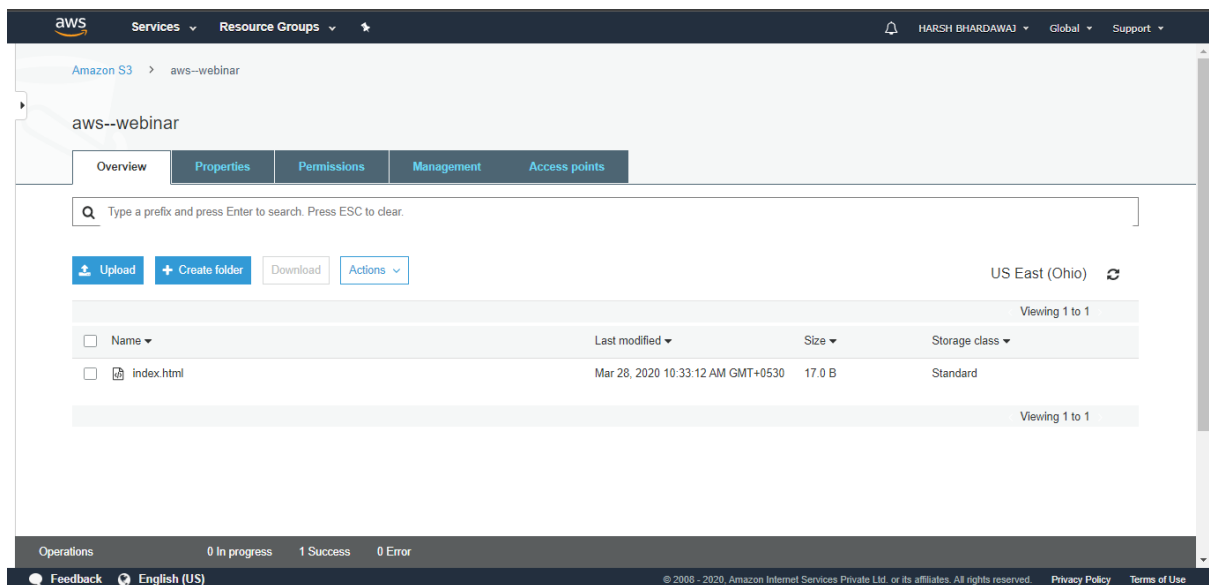
Step ii: Selecting a storage class:



Step iii: Review of upload:

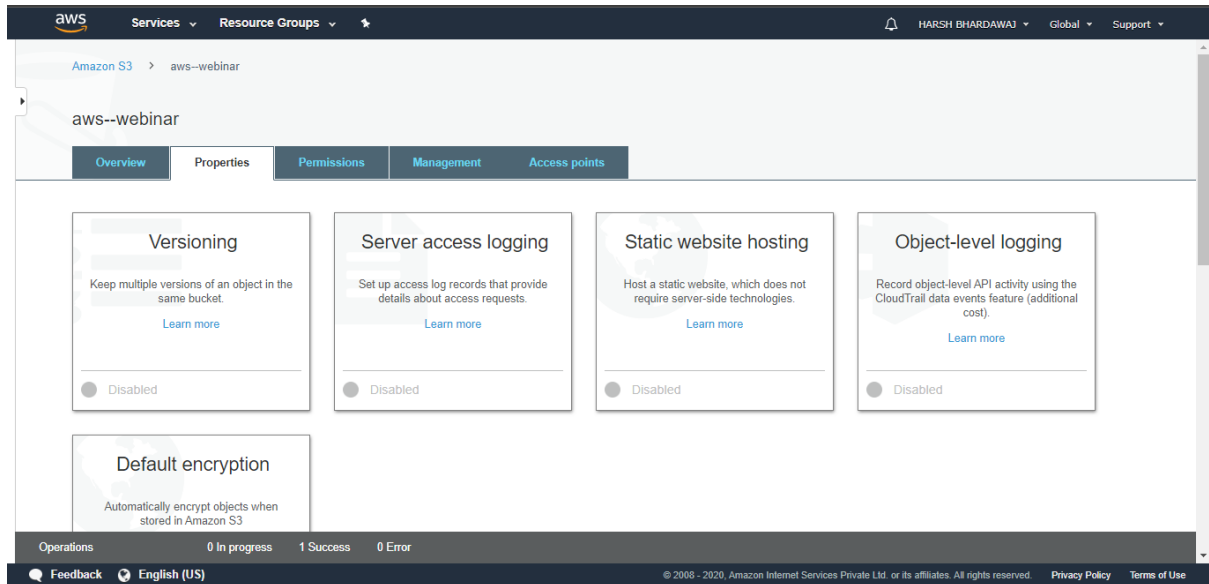


Step iv: After Uploading the Object:

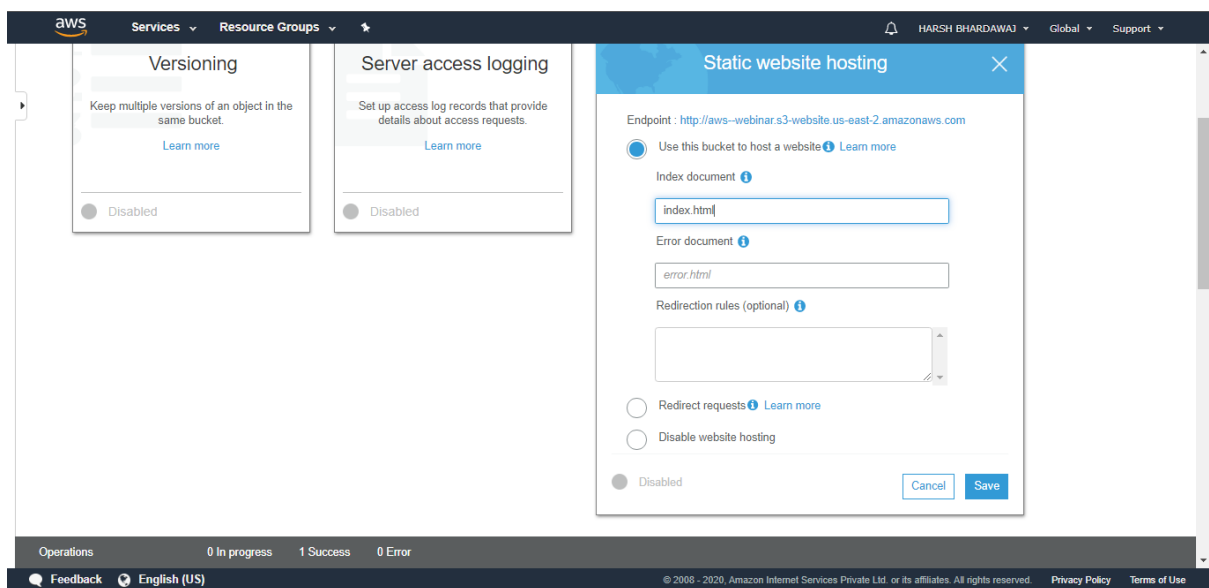


d) Enabling Static Website:

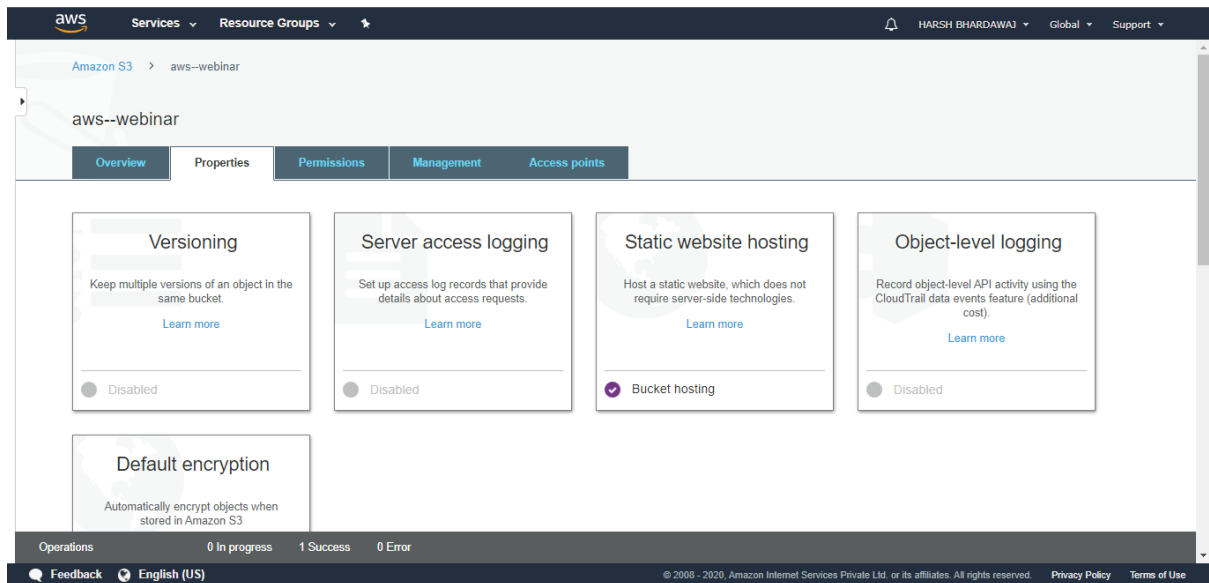
Step i: Properties of bucket created:



Step ii: Static website hosting:

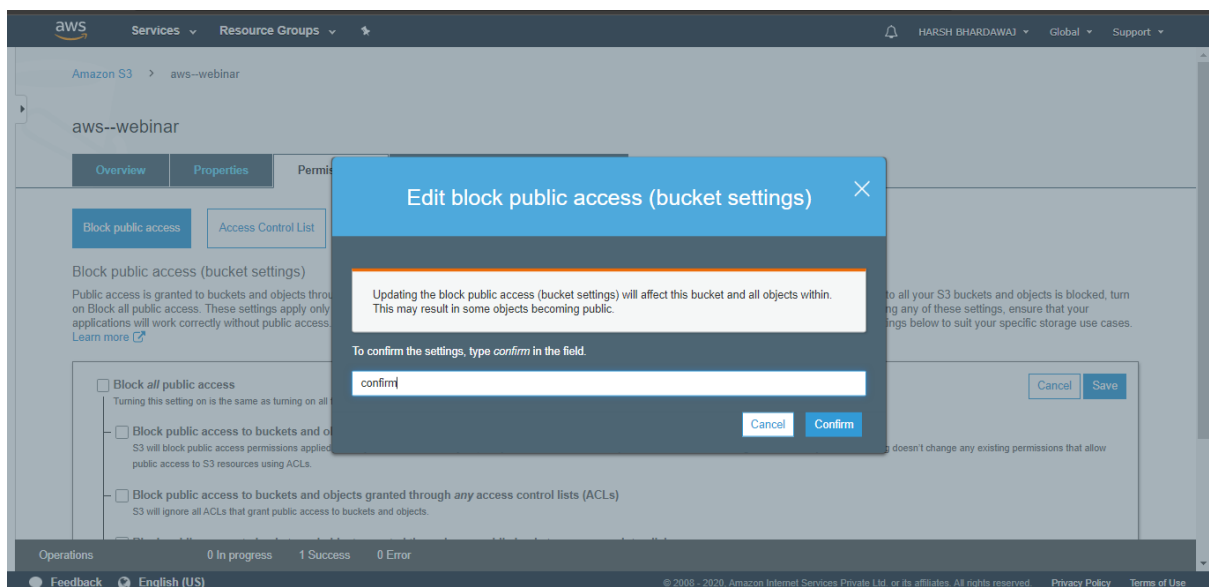


Step iii: Enabled Bucket hosting:

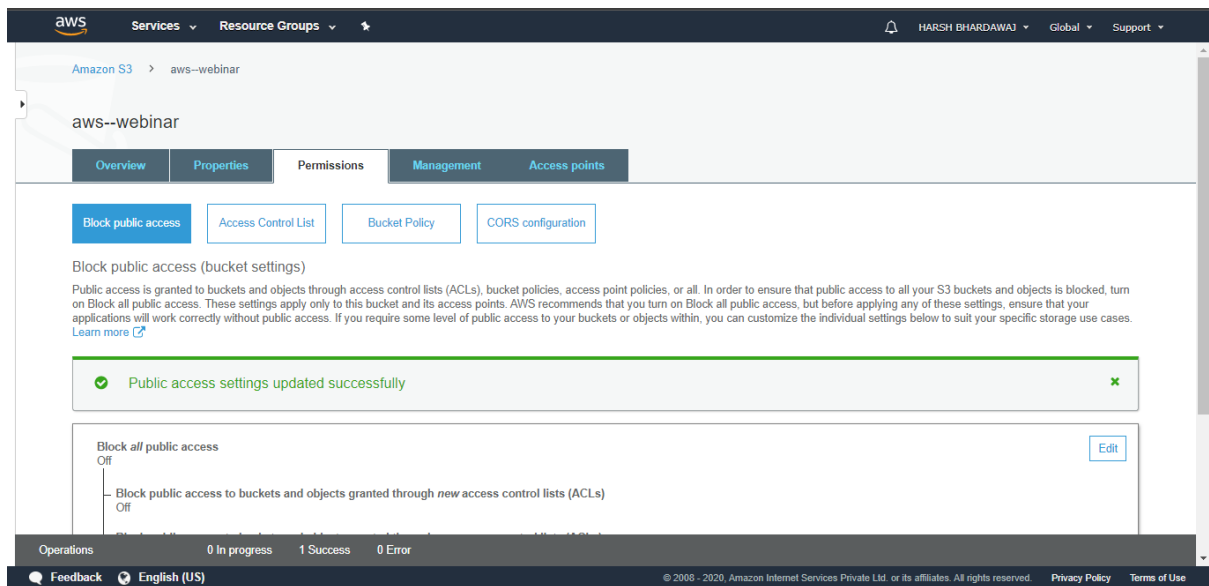


e) Making the Object Public:

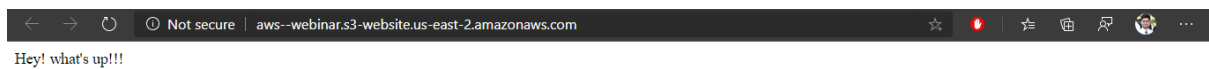
Step i: Disabling Public block:



Step ii: Successfully updated the public block permission:



f) Checking the S3 Link on the Browser:



Rekognition Screenshots:

a) Face Detect:

The screenshot shows the Amazon Rekognition 'Facial analysis' demo. The left sidebar lists various services, with 'Facial analysis' selected. The main area displays a group of Indian cricket players. Below the image, there are options to 'Choose a sample image' or 'Use your own image'. The 'Results' section on the right provides a detailed analysis of a selected face.

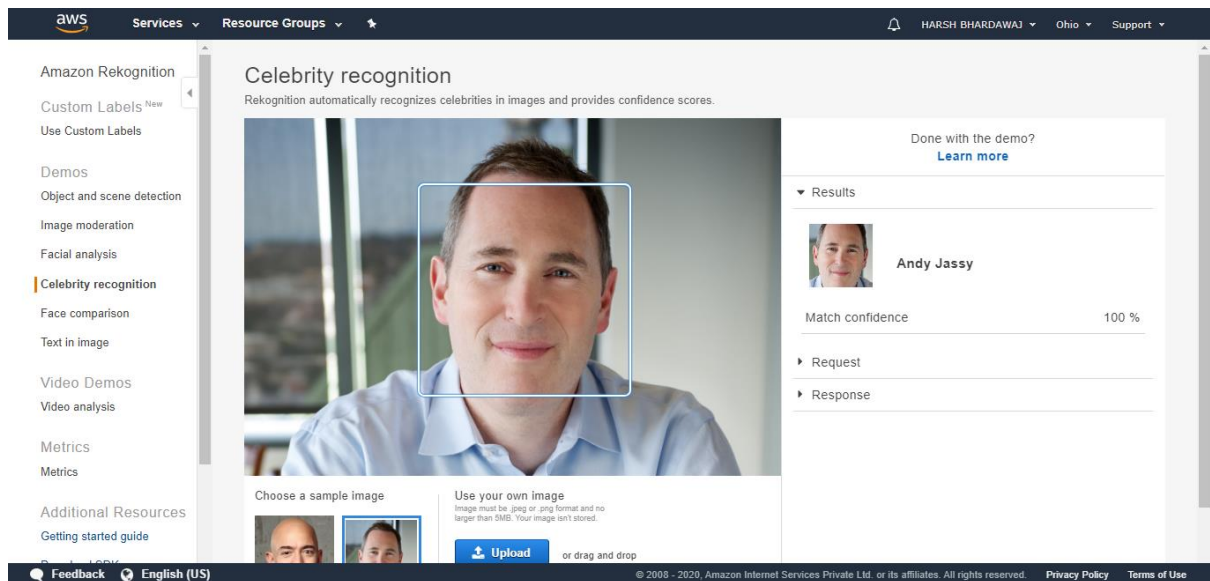
Attribute	Confidence Score
looks like a face	99.9 %
appears to be male	99.7 %
age range	25 - 39 years old
not smiling	97.7 %
appears to be calm	84.1 %
not wearing glasses	97.9 %

b) Face Compare:

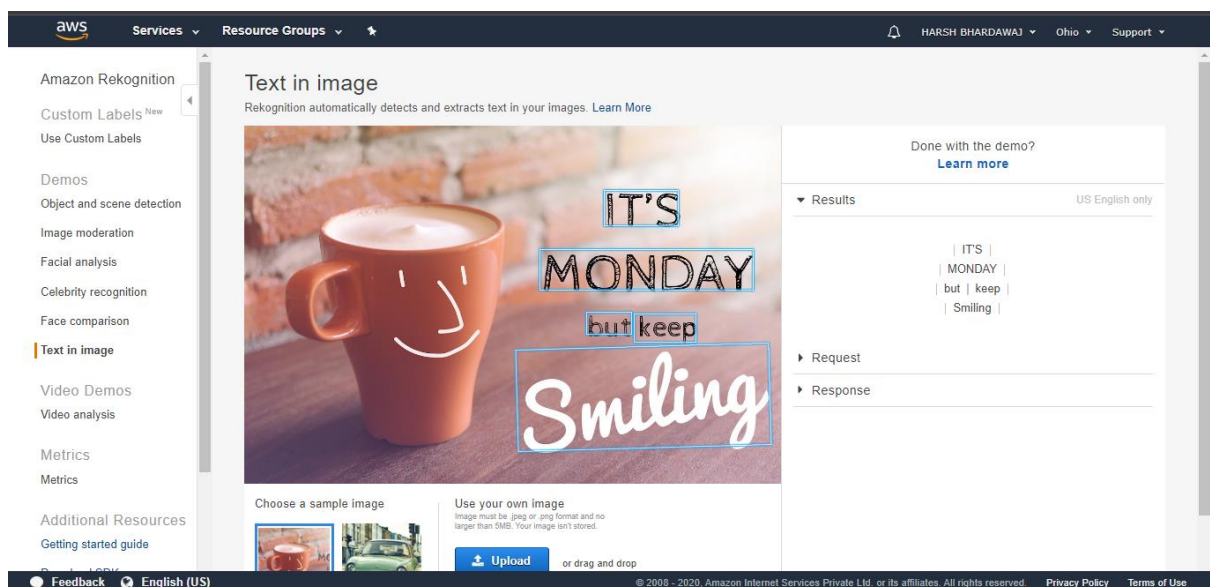
The screenshot shows the Amazon Rekognition 'Face comparison' demo. The left sidebar lists various services, with 'Face comparison' selected. The main area displays a 'Reference face' and 'Comparison faces'. Below the images, there are options to 'Choose a sample image'. The 'Results' section on the right shows the similarity percentage between the faces.

Comparison	Similarity
Reference face vs. Comparison face 1	99.9 %
Reference face vs. Comparison face 2	Not similar
Reference face vs. Comparison face 3	Not similar

c) Celebrity Recognition:



d) Text In Image:



EC2 & S3 Configuration Screenshots:

a) Installing AWS-SDK:

```
ec2-user@ip-172-31-17-190:/var/www/html/face
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 c
aching, request and response caching, and credentials caching)
aws/aws-sdk-php suggests installing monolog/monolog (Adds support for logging HT
TP requests and responses)
aws/aws-sdk-php suggests installing symfony/yaml (Eases the ability to write man
ifests for creating jobs in AWS Import/Export)
Package guzzle/guzzle is abandoned, you should avoid using it. Use guzzlehttp/gu
zlle instead.
Writing lock file
Generating autoload files
[ec2-user@ip-172-31-17-190 face]$ sudo php index.php
Image upload done... Here is the URL: https://aws--webinar.s3.us-east-2.amazonaws
[ec2-user@ip-172-31-17-190 face]$
```

b) Installing PHP:

```
ec2-user@ip-172-31-17-190:/var/www/html/face
17 lamp-mariadb10.2-php7.2 available \
   [ =10.2.10_7.2.0 =10.2.10_7.2.4 =10.2.10_7.2.5
     =10.2.10_7.2.8 =10.2.10_7.2.11 =10.2.10_7.2.13
     =10.2.10_7.2.14 =10.2.10_7.2.16 =10.2.10_7.2.17
     =10.2.10_7.2.19 =10.2.10_7.2.22 =10.2.10_7.2.23
     =10.2.10_7.2.24 =stable ]
18 libreoffice available \
   [ =5.0.6.2_15 =5.3.6.1 =stable ]
19 gimp available [ =2.8.22 ]
20 docker=latest enabled \
   [ =17.12.1 =18.03.1 =18.06.1 =18.09.9 =stable ]
21 mate-desktop.x available [ =1.19.0 =1.20.0 ]
22 GraphicsMagick1.3 available \
   [ =1.3.29 =1.3.32 =1.3.34 ]
23 tomcat8.5 available \
   [ =8.5.31 =8.5.32 =8.5.38 =8.5.40 =8.5.42 =8.5.50
     =stable ]
24 epel available [ =7.11 ]
25 testing available [ =1.0 ]
26 ecs available [ =stable ]
27 corretto8 available \
   [ =1.8.0_192 =1.8.0_202 =1.8.0_212 =1.8.0_222 =1.8.0_232
     =1.8.0_242 ]
28 firecracker available [ =0.11 =stable ]
29 go1.11.1 available \
   [ =1.11.3 =1.11.11 =1.11.13 =stable ]
30 squid4 available [ =4 ]
- php7.3 available \
   [ =7.3.2 =7.3.3 =7.3.4 =7.3.6 =7.3.8 =7.3.9 =7.3.10
     =7.3.11 =7.3.13 =stable ]
32 lustre2.10 available \
   [ =2.10.5 =2.10.8 =stable ]
33 java-openjdk11 available [ =11 =stable ]
34 lynis available [ =stable ]
35 kernel-ng available [ =stable ]
36 BCC available [ =0.x ]
37 mono available [ =5.x ]
38 nginx1 available [ =stable ]
39 ruby2.6 available [ =2.6 =stable ]
40 mock available [ =stable ]
41 postgresql11 available [ =11 =stable ]
- php7.4 available [ =stable ]
[ec2-user@ip-172-31-17-190 face]$
```

c) Index.php File Code:

```
ec2-user@ip-172-31-17-190:/var/www/html/face
sudo /sbin/mkswap /var/swap.1
sudo /sbin/swapoff /var/swap.1

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

<?
error_reporting(0);

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

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

$bucket = 'aws--webinar';
$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 --
```

d) Image Upload to S3:

Step i: Image Upload Successful:

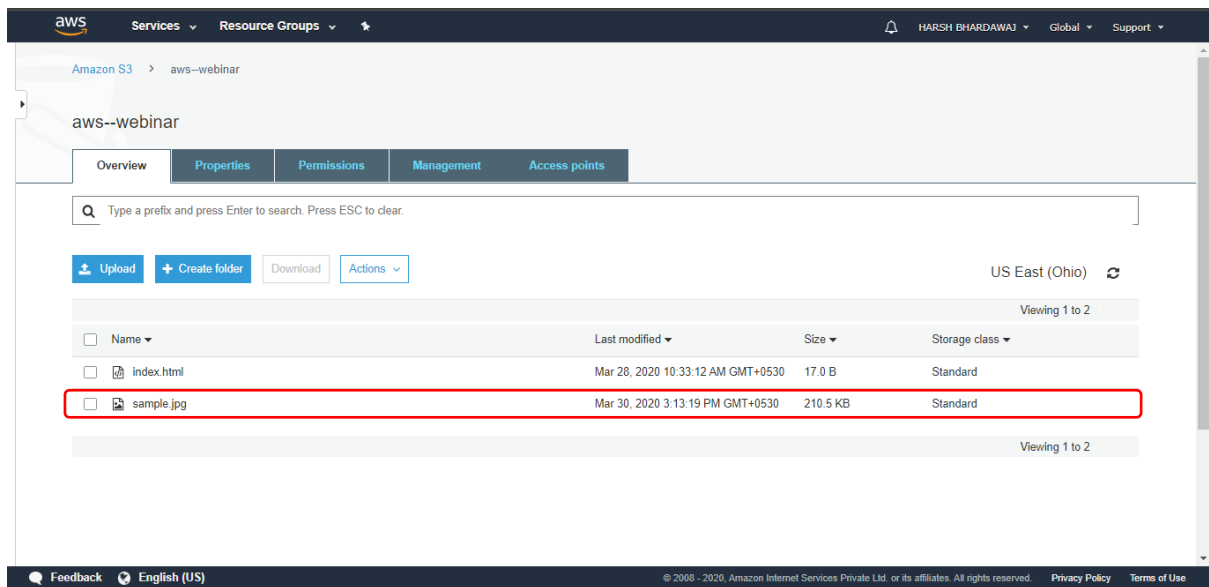
```
ec2-user@ip-172-31-17-190:/var/www/html/face

login as: ec2-user
Authenticating with public key "imported-openssh-key"
Last login: Mon Mar 30 09:30:48 2020 from 49.37.198.192

 _ _ | _ _ | _ )
 _ | ( _ _ /   Amazon Linux 2 AMI
 _ | \ _ _ | _ _ |

https://aws.amazon.com/amazon-linux-2/
1 package(s) needed for security, out of 7 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-17-190 ~]$ cd /var/www/html/face/
[ec2-user@ip-172-31-17-190 face]$ ls
composer.json  composer.lock  index.php  sample.jpg  vendor
[ec2-user@ip-172-31-17-190 face]$ sudo rm index.php
[ec2-user@ip-172-31-17-190 face]$ sudo vim index.php
[ec2-user@ip-172-31-17-190 face]$ sudo php index.php
Image upload done... Here is the URL: https://aws--webinar.s3.us-east-2.amazonaws.com
[ec2-user@ip-172-31-17-190 face]$
```

Step ii: Uploaded Image in S3:



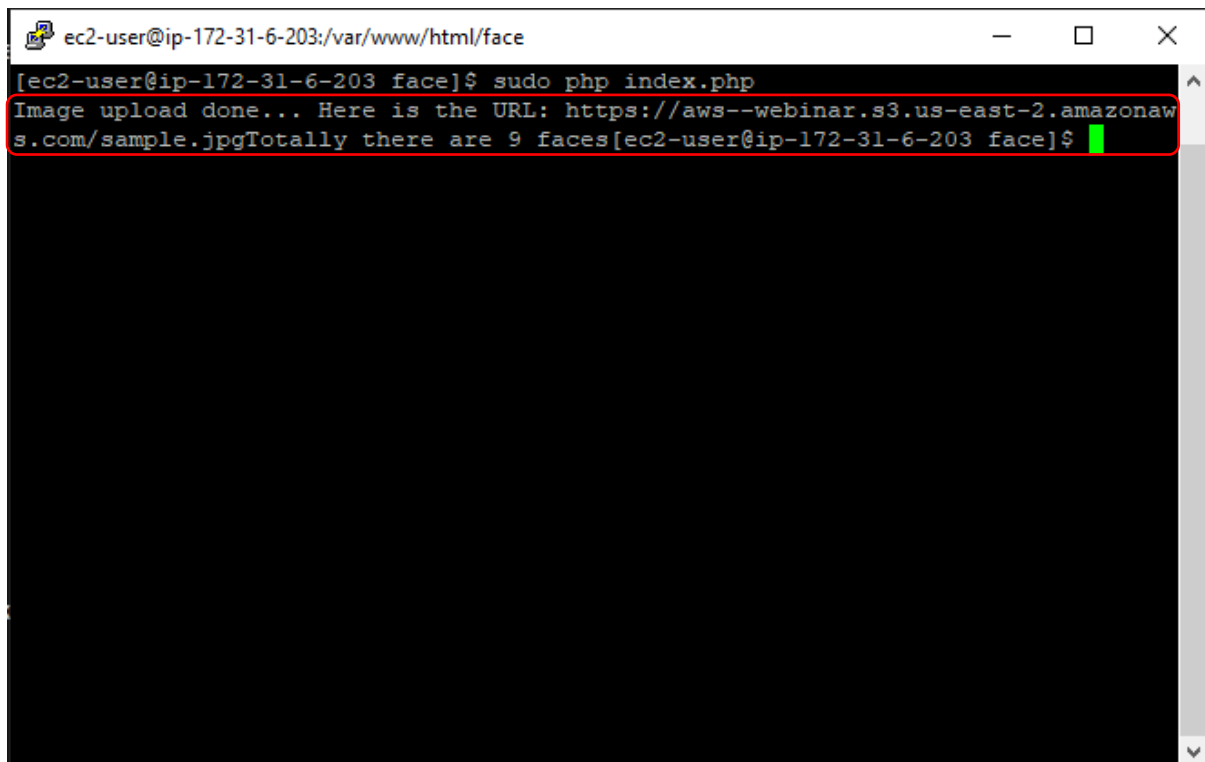
EC2 & Rekognition:

a) Face Detect Screenshot:

Step i: Feed in image:



Step ii: Successfully detected all the 9 faces from the uploaded image:

A terminal window titled 'ec2-user@ip-172-31-6-203:/var/www/html/face' with standard window controls. The terminal shows the command '[ec2-user@ip-172-31-6-203 face]\$ sudo php index.php' and its output: 'Image upload done... Here is the URL: https://aws--webinar.s3.us-east-2.amazonaws.com/sample.jpgTotally there are 9 faces'. The output line is highlighted with a red rectangular border. A green cursor is visible at the end of the command line.

```
ec2-user@ip-172-31-6-203:/var/www/html/face
[ec2-user@ip-172-31-6-203 face]$ sudo php index.php
Image upload done... Here is the URL: https://aws--webinar.s3.us-east-2.amazonaws.com/sample.jpgTotally there are 9 faces[ec2-user@ip-172-31-6-203 face]$
```

Conclusion:

Thus, using the Amazon Web Services we built the face recognition system based on the architecture as shown above. The Face detection is done and all the results are depicted in the report.