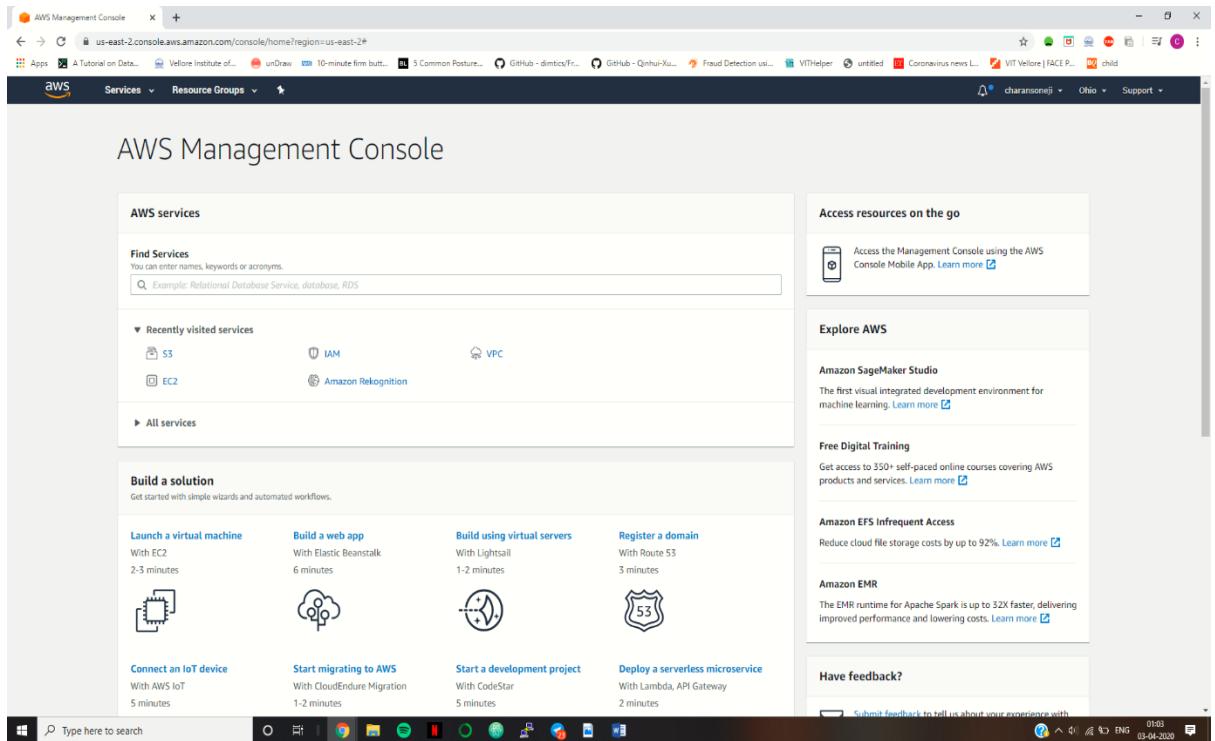


**Name:** Charan Lalchand Soneji

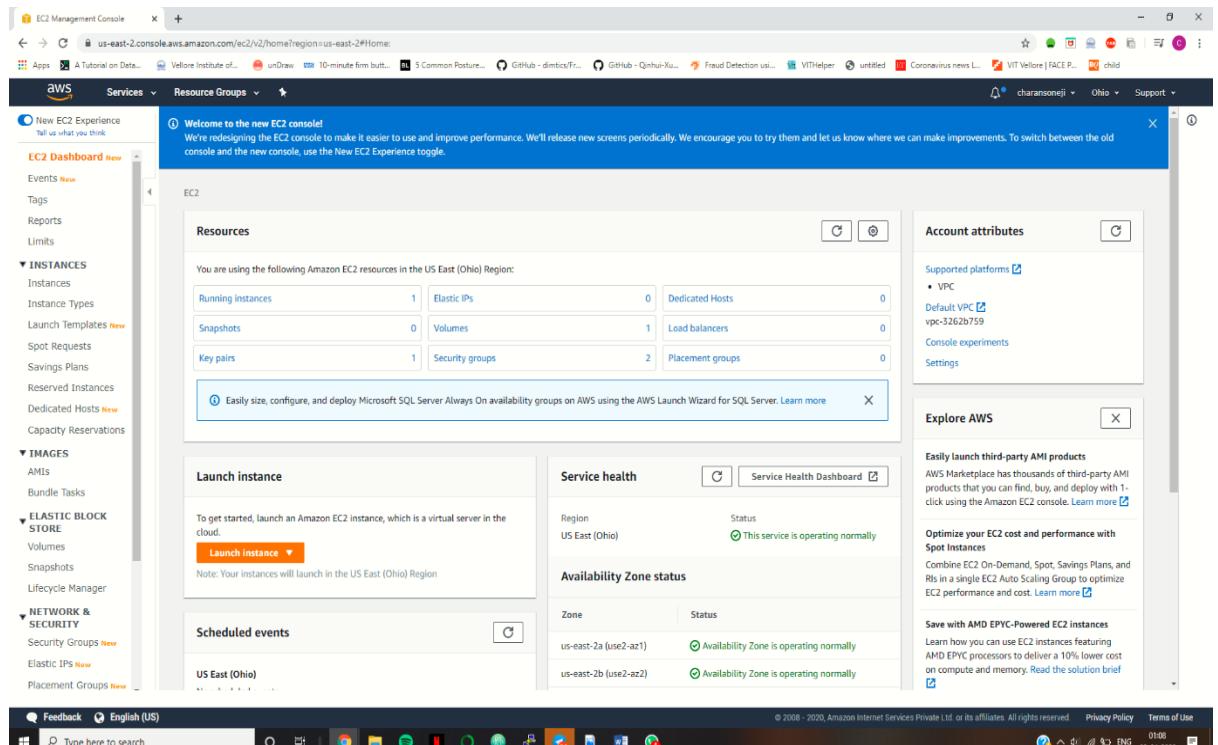
## AWS

### 1. AWS Login Screen with username

Login screen after logging in



### 2. EC2 Dashboard



### 3. S3 Dashboard

The screenshot shows the AWS S3 Management Console. The left sidebar has a 'Buckets' section with links for 'Batch operations', 'Access analyzer for S3', 'Block public access (account settings)', and 'Feature spotlight'. The main content area is titled 'Amazon S3' and shows a table for 'Buckets (1)'. The table has columns for Name, Region, Access, and Bucket created. One bucket named 'aws-charan.s90' is listed, created on '2020-04-01T18:06:02.000Z'. There are buttons for 'Copy ARN', 'Empty', 'Delete', and 'Create bucket'.

### 4. Rekognition Dashboard

The screenshot shows the AWS Rekognition Console. The left sidebar has sections for 'Amazon Rekognition', 'Custom Labels', 'Demos', 'Image moderation', 'Facial analysis', 'Celebrity recognition', 'Face comparison', 'Text in image', 'Video Demos', 'Video analysis', 'Metrics', 'Additional Resources', 'Feature Spotlight', and 'Learning Content'. The main content area features a large blue background image of a network graph. It includes a 'Try Demo' button, a 'Download SDKs' link, and three icons: a stack of three squares labeled 'Easily Integrate Powerful Visual Analysis into Your App', a circuit board labeled 'Continuously Learning', and two interlocking gears labeled 'Integrated with AWS Services'. The 'Integrated with AWS Services' section describes how Rekognition integrates with other AWS services like S3 and Lambda for scalable video analysis.

## 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

My AMIs

AWS Marketplace

Community AMIs

Free tier only

Amazon Linux (Free tier eligible)

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0e01ce4ee18447327 (64-bit x86) / ami-03201f374ab56a26e (64-bit Arm)

Amazon Linux comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 210, GCC 7.3, libc 2.26, Binutils 2.29.1, and the latest software packages through extras.

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.

Select  
64-bit (x86)

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

Select  
64-bit (x86)  
64-bit (Arm)

SUSE Linux Enterprise Server 15 SP1 (HVM), SSD Volume Type - ami-04c5bab51cc146925 (64-bit x86) / ami-02e73902018018171 (64-bit Arm)

SUSE Linux Enterprise Server 15 Service Pack 1 (HVM), EBS General Purpose (SSD) Volume Type. Public Cloud, Advanced Systems Management, Web and Scripting, and Legacy modules enabled.

Select  
64-bit (x86)  
64-bit (Arm)

Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-0fc20dd1da406780b (64-bit x86) / ami-0959e8feedfa156bf (64-bit Arm)

Ubuntu Server 18.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Select  
64-bit (x86)  
64-bit (Arm)

Are you launching a database instance? Try Amazon RDS.

Amazon Relational Database Service (RDS) makes it easy to set up, operate, and scale your database on AWS by automating time-consuming database management tasks. With RDS, you can easily deploy Amazon Aurora, MariaDB,

### 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 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
General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
General purpose	t2.micro (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
General purpose	t2.xlarge	4	16	EBS only	-	Moderate	Yes
General purpose	t2.2xlarge	8	32	EBS only	-	Moderate	Yes
General purpose	t3a.nano	2	0.5	EBS only	Yes	Up to 5 Gigabit	Yes
General purpose	t3a.micro	2	1	EBS only	Yes	Up to 5 Gigabit	Yes
General purpose	t3a.small	2	2	EBS only	Yes	Up to 5 Gigabit	Yes
General purpose	t3a.medium	2	4	EBS only	Yes	Up to 5 Gigabit	Yes
General purpose	t3a.large	2	8	EBS only	Yes	Up to 5 Gigabit	Yes

Cancel Previous Review and Launch Next: Configure Instance Details

Feedback English (US) Type here to search © 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use 01/24 03-04-2020

**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	<input type="checkbox"/> Request Spot instances	
Network	vpc-3262b759 (default)	<input type="button"/> Create new VPC
Subnet	No preference (default subnet in any Availability Zone)	<input type="button"/> Create new subnet
Auto-assign Public IP	<input type="checkbox"/> Use subnet setting (Enable)	
Placement group	<input type="checkbox"/> Add instance to placement group	
Capacity Reservation	Open	<input type="button"/> Create new Capacity Reservation
IAM role	<input type="checkbox"/> None	
Shutdown behavior	<input type="checkbox"/> Stop	
Stop - Hibernate behavior	<input type="checkbox"/> Enable hibernation as an additional stop behavior	
Enable termination protection	<input type="checkbox"/> Protect against accidental termination	
Monitoring	<input type="checkbox"/> Enable CloudWatch detailed monitoring Additional charges apply.	
Tenancy	<input type="checkbox"/> Shared - Run a shared hardware instance Additional charges will apply for dedicated tenancy.	
Elastic Inference	<input type="checkbox"/> Add an Elastic Inference accelerator Additional charges apply.	
T2/T3 Unlimited	<input type="checkbox"/> Enable Additional charges may apply	

**Review and Launch**

### 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.](#)

**Review and Launch**

## 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: launch-wizard-2

Description: launch-wizard-2 created 2020-04-03T01:29:28.317+04:00

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.

## 5. Key-Pair Download

Step 7: Review Instance Launch

AMI Details

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0e01ce4ee1844732

Instance Type

Instance Type	ECUs	vCPUs	Memory (GiB)	Storage (GiB)
t2.micro	Variable	1	1	8

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.

Create a new key pair

Key pair name: aws-facereko-2

You have to download the **private key file** (.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

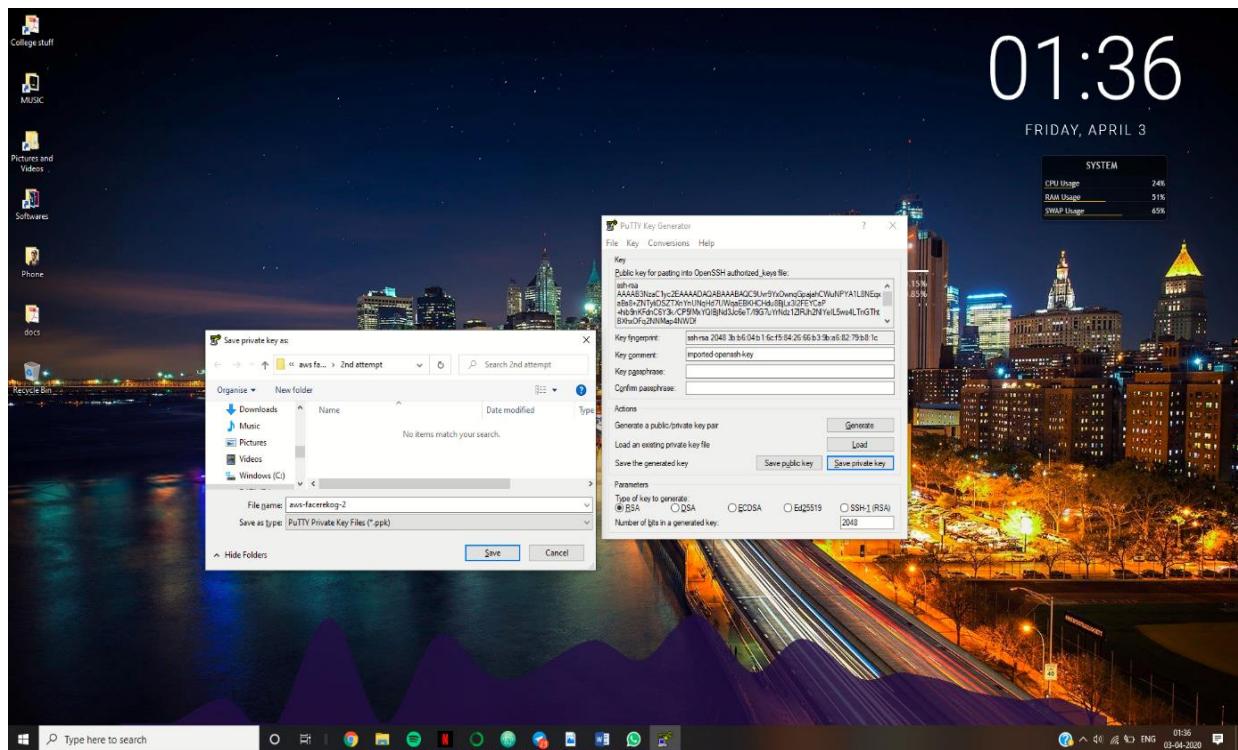
Save As

File name: aws-facereko-2

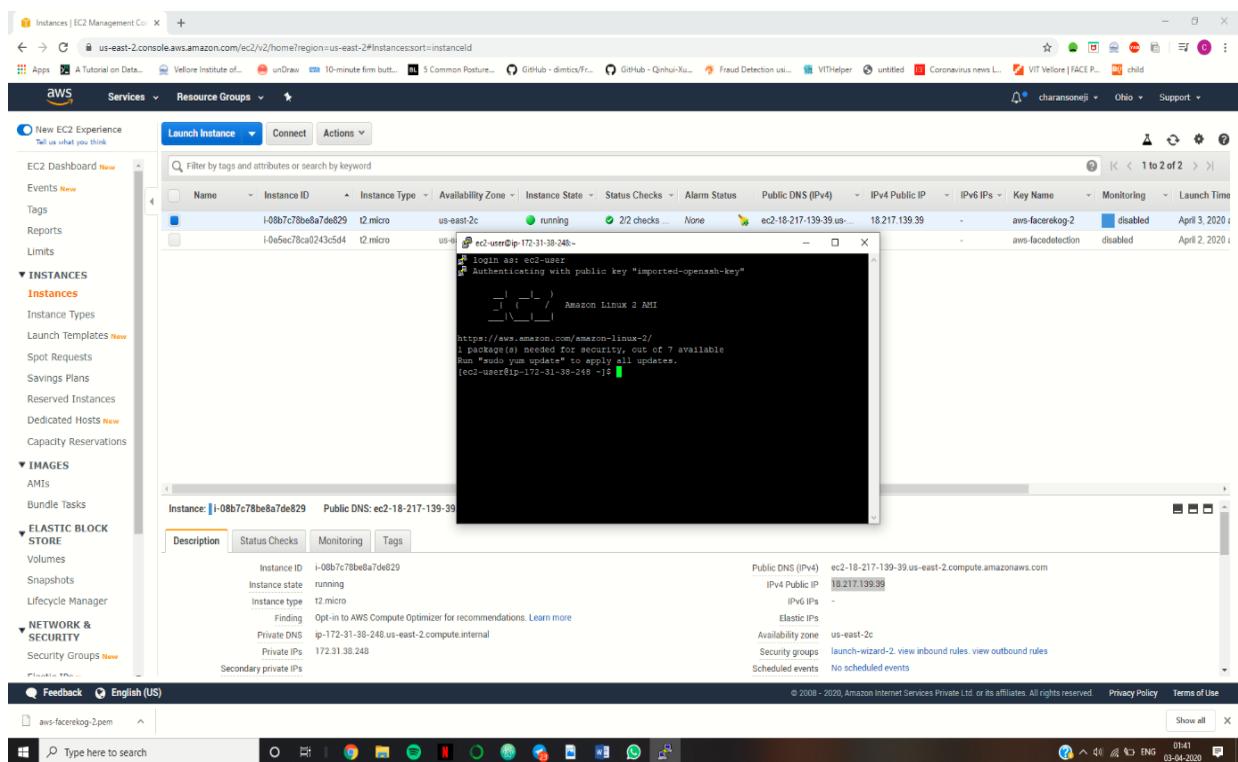
Save as type: PEM File

Cancel Launch Instances

## 6. PuTTYgen conversion from pem to ppk



## 7. Logged in EC2 black screen



## S3

### 1. Creating a bucket

The screenshot shows the 'Create bucket' wizard in the AWS S3 Management Console. The 'General configuration' step is selected. A bucket named 'aws-charan.s90' is being created in the 'US East (Ohio) us-east-2' region. Under 'Bucket settings for Block Public Access', the 'Block all public access' checkbox is checked. The status bar at the bottom indicates the session is from March 4, 2020, at 01:46 AM.

### 2. Uploading an Object

The screenshot shows the 'Upload' wizard in the AWS S3 Management Console. The 'aws-charan.s90' bucket is selected. The 'Select files' step is active, showing a file selection dialog. The dialog lists files in the 'aws-facerecog-2' folder, including 'aws-facerecog-2.perm', 'aws-facerecog-2', and 'index'. The 'index' file is selected. The status bar at the bottom indicates the session is from March 4, 2020, at 01:49 AM.

The screenshot shows the AWS S3 Management Console interface. At the top, there's a navigation bar with tabs for Services, Resource Groups, and a user dropdown. Below that, the main area shows a bucket named 'aws-charan.s90'. A sub-menu bar at the top of this area includes 'Overview', 'Properties' (which is selected), 'Permissions', 'Management', and 'Access points'. A search bar below the sub-menu contains the placeholder text 'Type a prefix and press Enter to search. Press ESC to clear.' Underneath, there are buttons for 'Upload', '+ Create folder', 'Download', and 'Actions'. To the right, it says 'US East (Ohio)' and 'Viewing 1 to 1'. The main content area lists the single object 'index.html' with its details: Name (index.html), Last modified (Apr 3, 2020 1:51:23 AM GMT+0400), Size (45.0 B), and Storage class (Standard). At the bottom, there's a summary of operations: 0 In progress, 1 Success, 0 Error.

### 3. Enabling Static Website

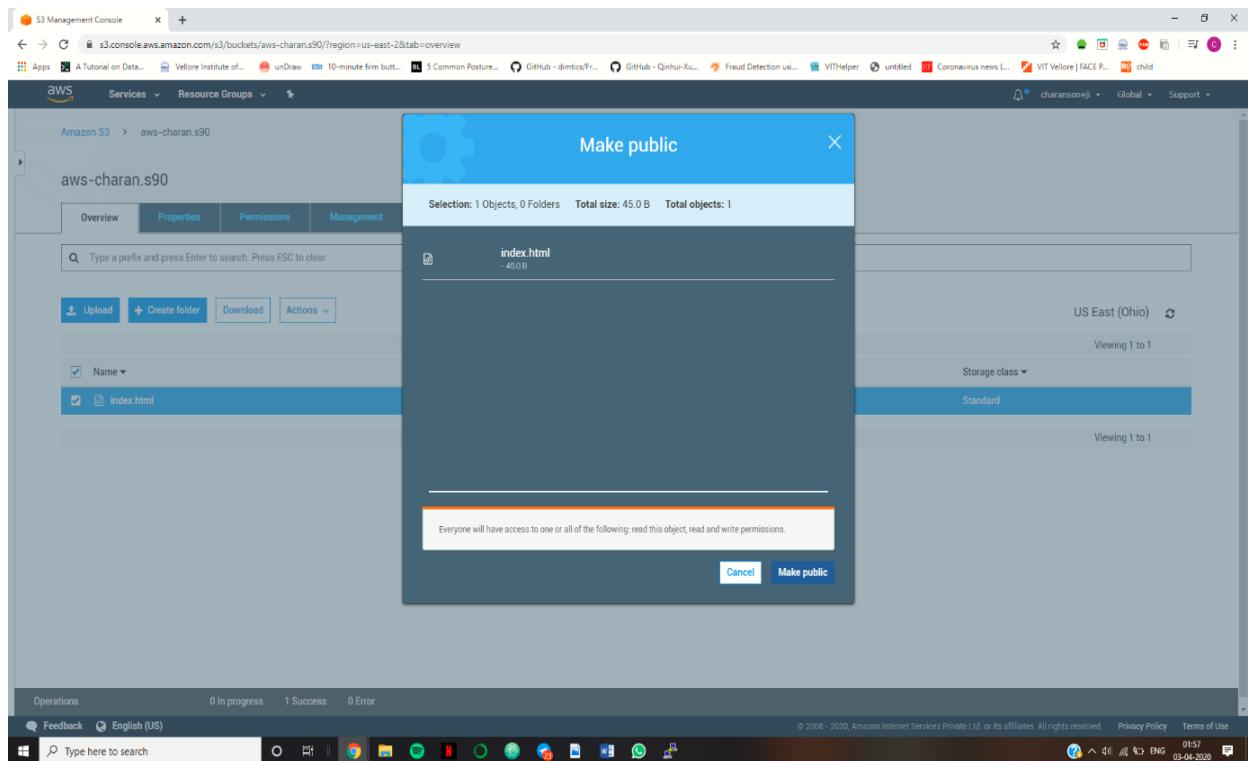
The screenshot shows the AWS S3 Management Console Properties tab for the 'aws-charan.s90' bucket. The 'Static website hosting' section is the active tab, displaying the endpoint 'http://aws-charan.s90.s3-website-us-east-2.amazonaws.com'. It includes fields for 'Index document' (set to 'index.html') and 'Error document' (set to 'error.html'). There are also options for 'Redirection rules (optional)' and 'Disable website hosting'. Other tabs visible include 'Versioning', 'Server access logging', and 'Object-level logging', each with their own specific configurations. At the bottom, there's a summary of operations: 0 In progress, 1 Success, 0 Error.

## Enabled tick sign on Static Web hosting

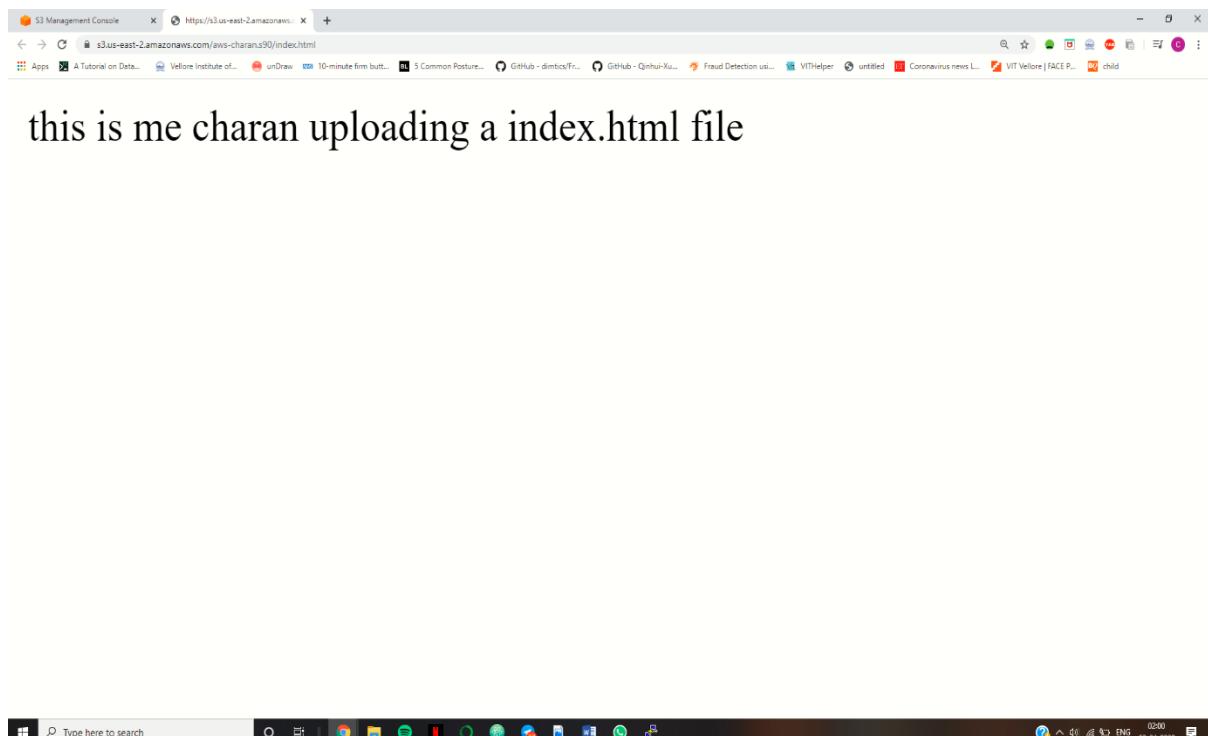
The screenshot shows the AWS S3 Management Console for the bucket 'aws-charan.s90'. The 'Properties' tab is selected. Under 'Static website hosting', the 'Bucket hosting' option is highlighted with a purple circle and the text 'Enabled'. Other options like 'Versioning', 'Server access logging', 'Object-level logging', and 'Default encryption' are also listed but not enabled.

## 4. Making the Object Public

The screenshot shows the AWS S3 Management Console for the bucket 'aws-charan.s90'. The 'Properties' tab is selected. A modal dialog titled 'Edit block public access (bucket settings)' is open. It contains several checkboxes under 'Block public access (bucket settings)'. One checkbox, 'Block all public access', is checked and highlighted with a blue border. A confirmation message at the bottom of the dialog says: 'Updating the block public access (bucket settings) will affect this bucket and all objects within. This may result in some objects becoming public.' There is a 'confirm' input field and 'Cancel' and 'Confirm' buttons. The background shows other tabs like 'Access Control List' and 'Bucket Policy'.

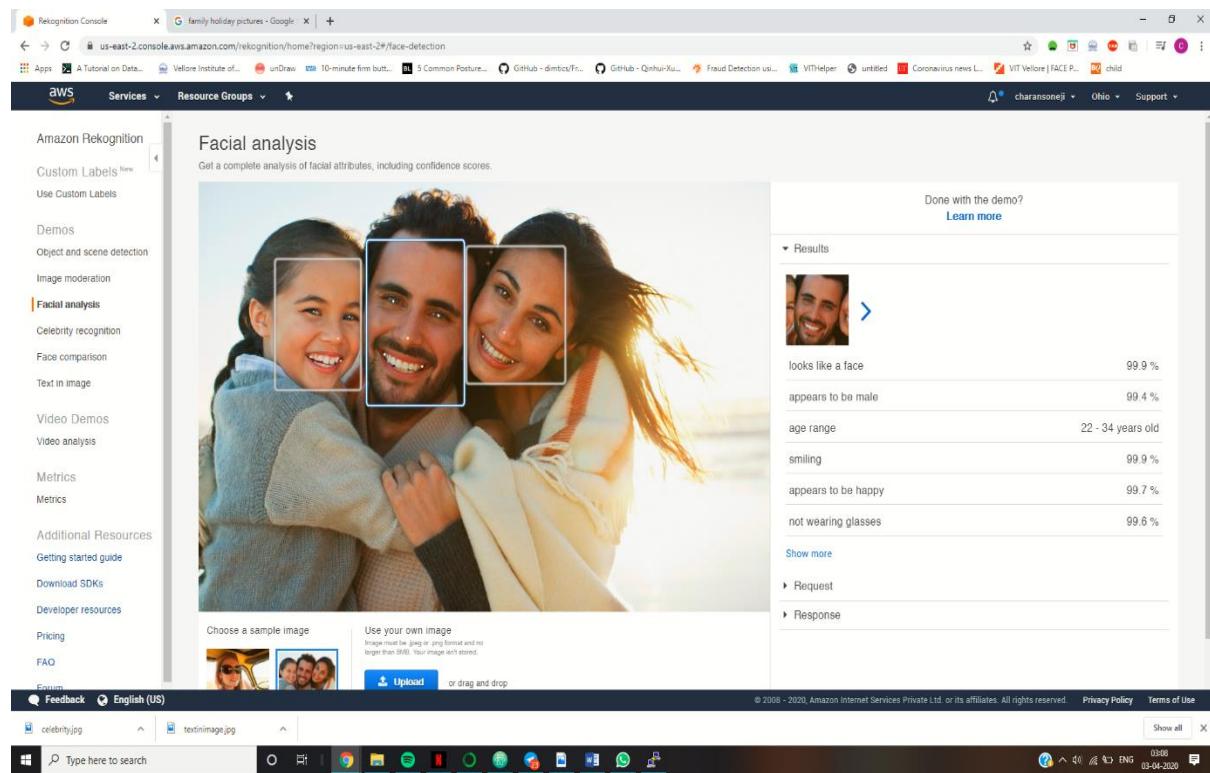


## 5. Check the S3 link in browser



## Rekognition

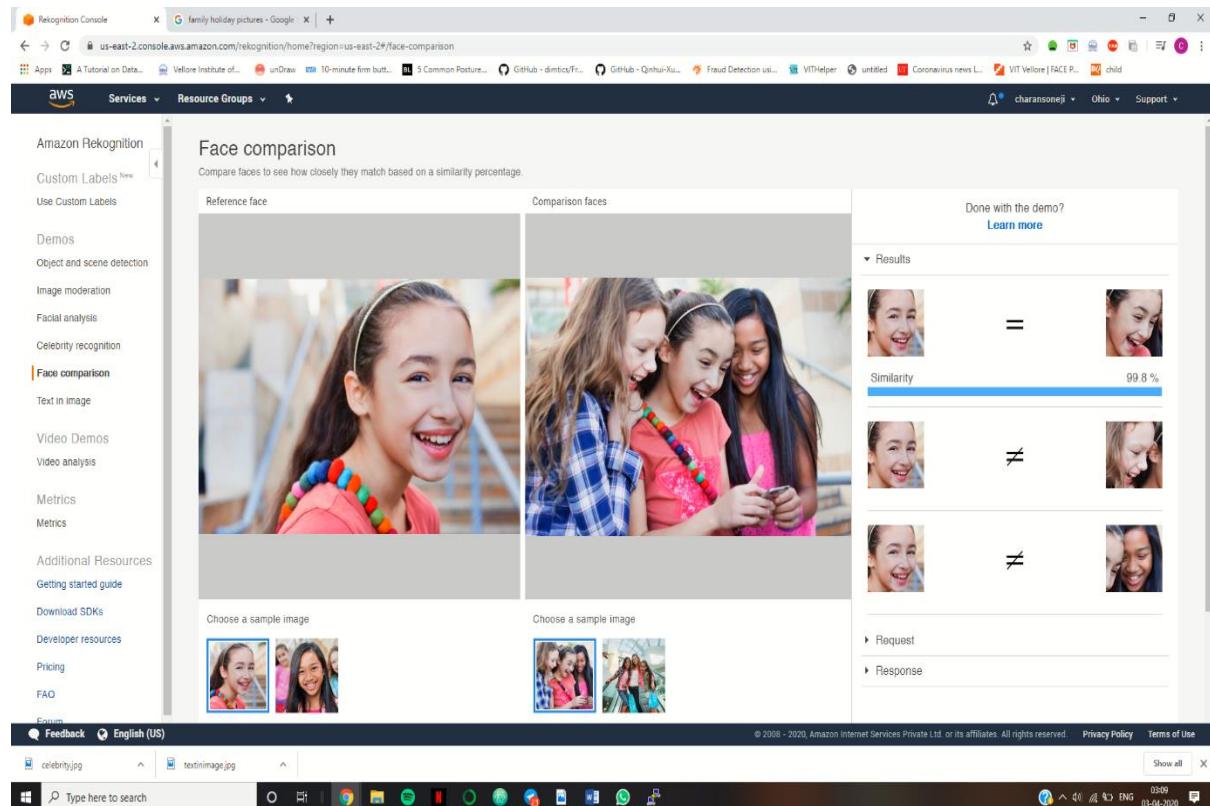
### 1. Face Detect



The screenshot shows the AWS Rekognition Facial analysis interface. On the left, a sidebar lists various services like Custom Labels, Demos, and Metrics. The main area displays a photo of a smiling family. Three faces are highlighted with bounding boxes. To the right, a results panel shows the following analysis:

Attribute	Score (%)
looks like a face	99.9 %
appears to be male	99.4 %
age range	22 - 34 years old
smiling	99.9 %
appears to be happy	99.7 %
not wearing glasses	99.6 %

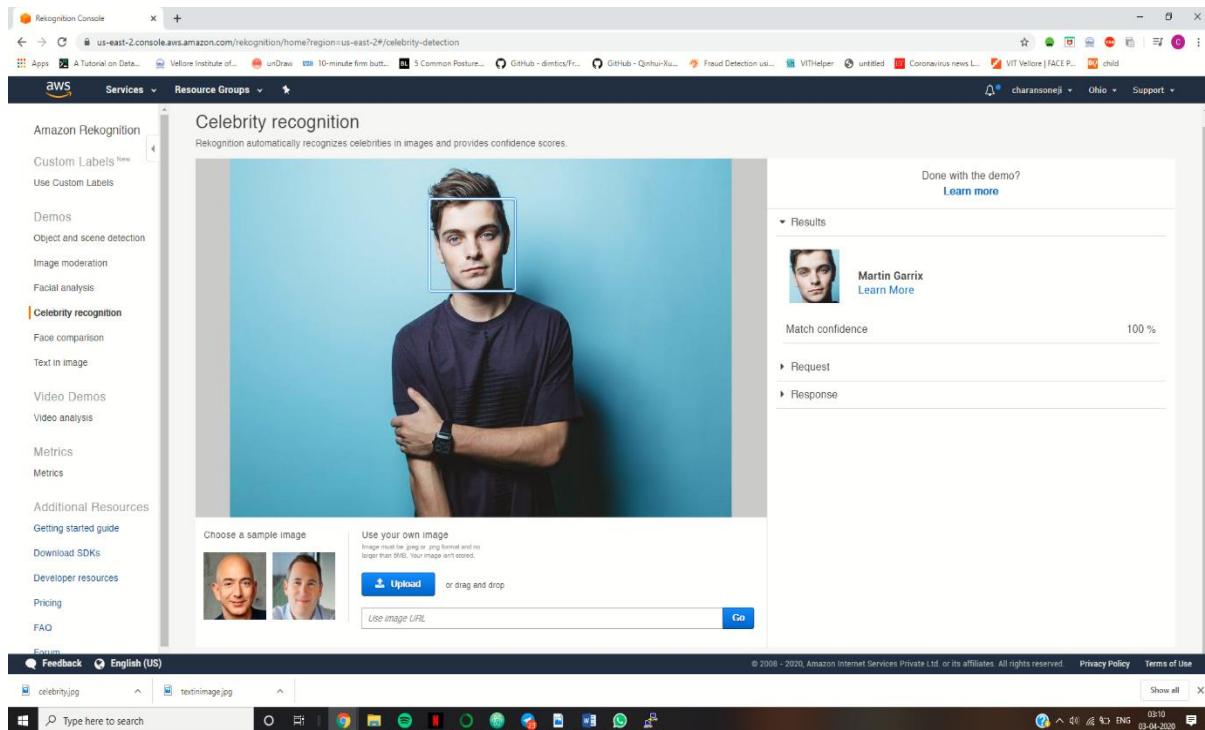
### 2. Face Compare



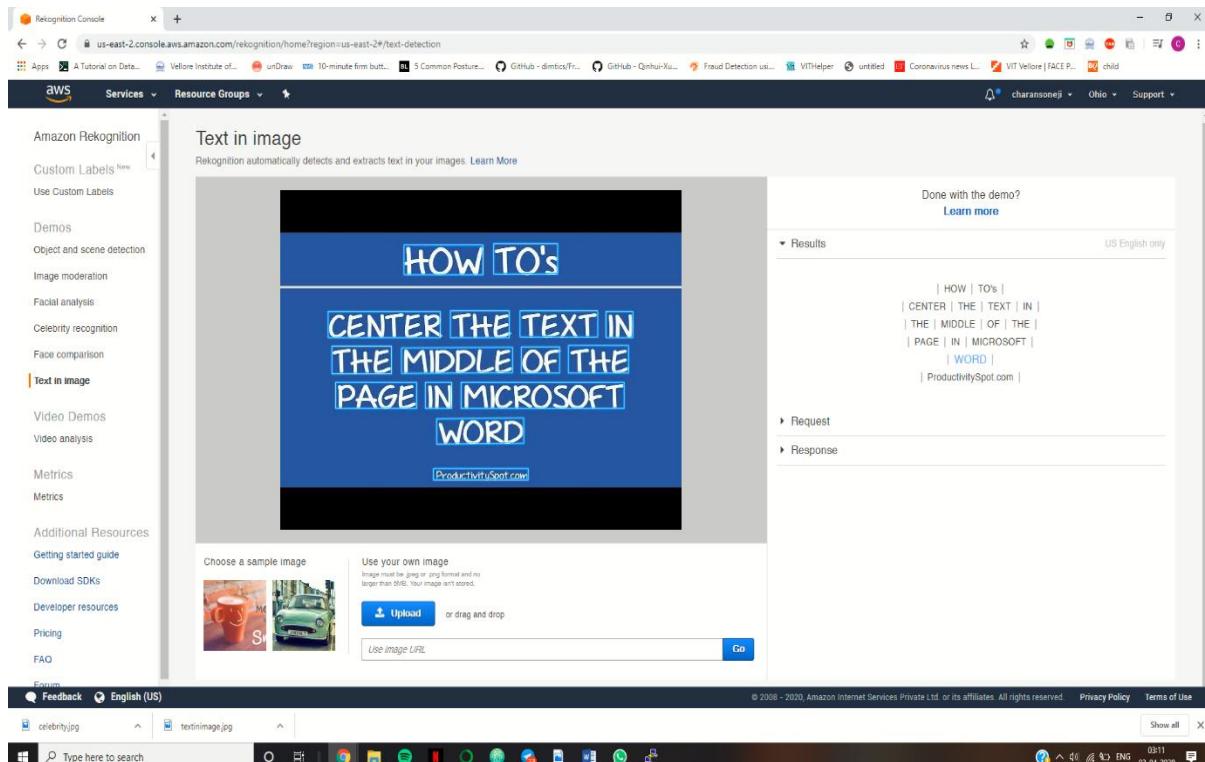
The screenshot shows the AWS Rekognition Face comparison interface. On the left, a sidebar lists various services like Custom Labels, Demos, and Metrics. The main area shows two reference faces: a young girl in a pink top and another girl in a plaid shirt. Below them are two comparison images. The results panel on the right shows the following similarity scores:

Comparison	Similarity (%)
Reference face vs. Comparison 1	99.8 %
Reference face vs. Comparison 2	99.8 %
Reference face vs. Comparison 3	99.8 %

### 3. Celebrity Recognition

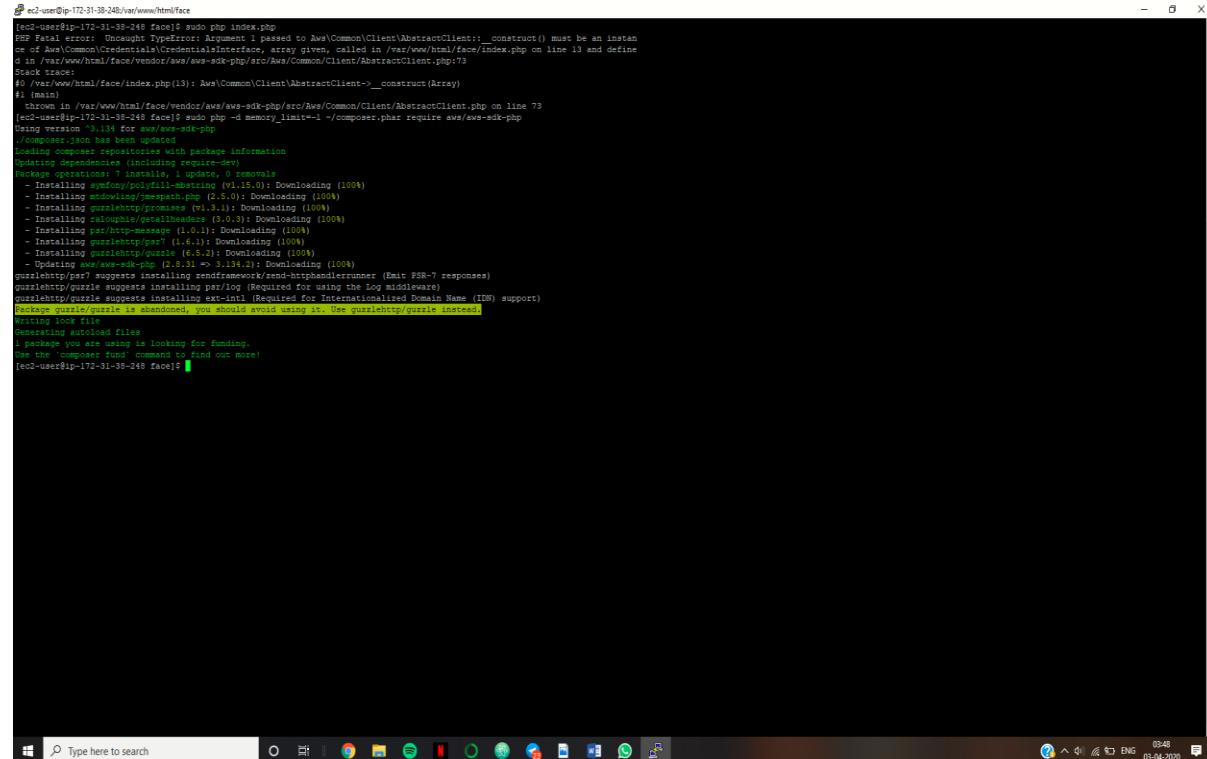


### 4. Text in Image



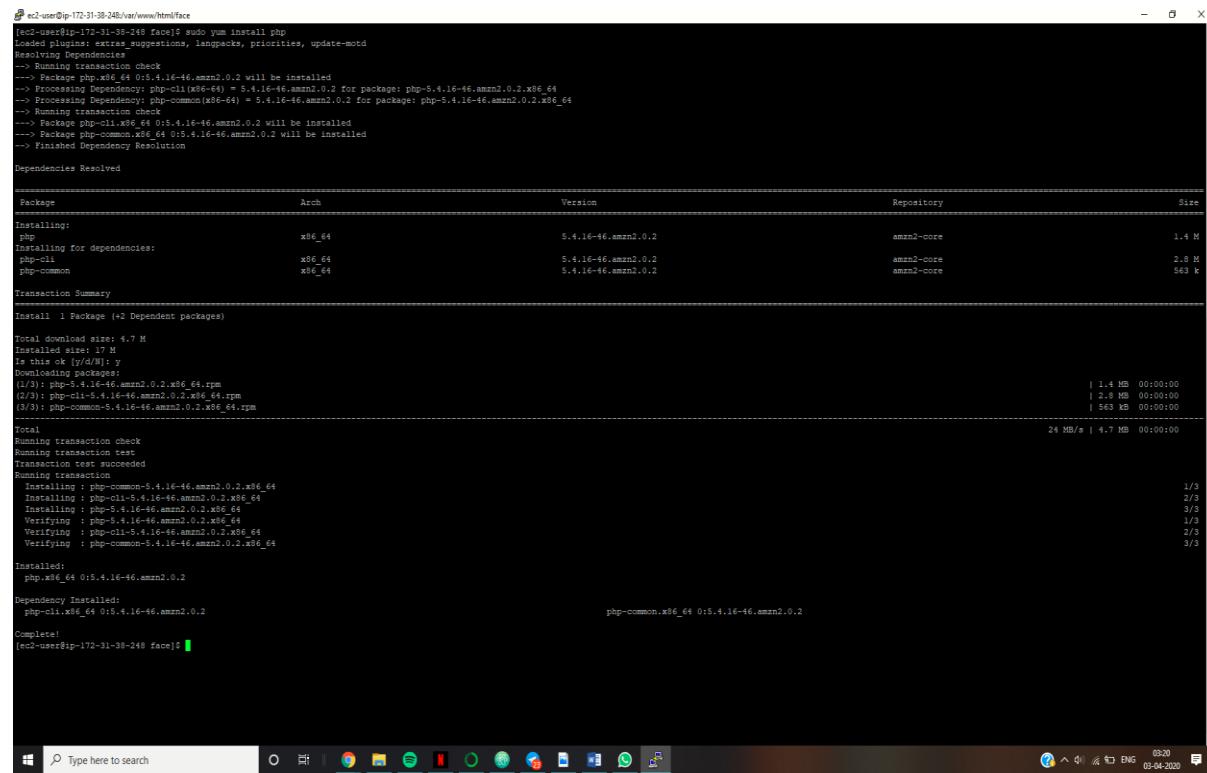
## Screenshots needed for EC2 and S3

### 1. Installing aws-sdk



```
[ec2-user@ip-172-31-38-248 ~]$ composer require aws/aws-sdk-php
[ec2-user@ip-172-31-38-248 face]$ sudo php index.php
PHP Fatal error:  Uncaught TypeError: Argument 1 passed to Aws\Common\Client\AbstractClient::__construct() must be an instance of Aws\Common\Credentials\Interface, array given. Client in /var/www/html/face/index.php on line 13 and defined in /var/www/html/face/vendor/aws/aws-sdk-php/src/Aws/Common/Client/AbstractClient.php:73
Stack trace:
#0 /var/www/html/face/index.php(13): Aws\Common\Client\AbstractClient::__construct(Array)
#1 {main}
    thrown in /var/www/html/face/vendor/aws/aws-sdk-php/src/Aws/Common/Client/AbstractClient.php on line 73
[ec2-user@ip-172-31-38-248 face]$ sudo php -d memory_limit=1 /composer.phar require aws/aws-sdk-php
Using version ^3.11.4 for aws/aws-sdk-php
./composer.json has been updated
Loading composer repositories with package information
Updating dependencies (including require-dev)
  - No packages need updating, 0 installs, 0 removals
  - Installing symfony/polyfill-mbstring (v1.15.0): Downloading (100%)
  - Installing ext-sodium/jmespath.php (2.5.0): Downloading (100%)
  - Installing guzzlehttp/promises (v1.3.1): Downloading (100%)
  - Installing guzzlehttp/guzzle (v6.5.3): Downloading (100%)
  - Installing guzzlehttp/psr7 (1.6.1): Downloading (100%)
  - Installing guzzlehttp/guzzle (6.5.3): Downloading (100%)
  - Updating aws/aws-sdk-php (2.8.31 => 3.13.4): Downloading (100%)
guzzlehttp/psr7 suggests installing zendframework/zend-stdlib|zend-dicontainer (Emits PSR-7 responses)
guzzlehttp/psr7 suggests installing psr/http-message (For handling the Log middleware)
guzzlehttp/guzzle suggests installing ext-intl (Required for Internationalized Domain Name (IDN) support)
Package guzzle/guzzle is abandoned, you should avoid using it. Use guzzlehttp/guzzle instead.
Writing lock file
Generating autoload files
package aws/aws-sdk-php is looking for funding.
Use the "composer fund" command to find out more!
[ec2-user@ip-172-31-38-248 face]$
```

### 2. Installing php



```
[ec2-user@ip-172-31-38-248 ~]$ sudo yum install php
[ec2-user@ip-172-31-38-248 face]$ 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-5.4.16-46.amzn2.0.2.x86_64
--> Processing Dependency: php-common(x86-64) = 5.4.16-46.amzn2.0.2 for package: php-5.4.16-46.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
--> 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:
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 (+2 Dependent packages)

Total download size: 4.7 M
Total download size: 4.7 M
Is this ok [y/d/N]: y
Downloading packages:
(1/3): php-5.4.16-46.amzn2.0.2.x86_64.rpm                                | 1.4 MB  00:00:00
(2/3): php-cli-5.4.16-46.amzn2.0.2.x86_64.rpm                               | 2.8 MB  00:00:00
(3/3): php-common-5.4.16-46.amzn2.0.2.x86_64.rpm                            | 563 KB  00:00:00
                                                               24 MB/s | 4.7 MB  00:00:00

Total
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : php-common-5.4.16-46.amzn2.0.2.x86_64
  Installing : php-cli-5.4.16-46.amzn2.0.2.x86_64
  Installing : php-5.4.16-46.amzn2.0.2.x86_64
  Verifying  : php-common-5.4.16-46.amzn2.0.2.x86_64
  Verifying  : php-cli-5.4.16-46.amzn2.0.2.x86_64
  Verifying  : php-5.4.16-46.amzn2.0.2.x86_64
  Installed:
    php.x86_64 0:5.4.16-46.amzn2.0.2
  Dependency Installed:
    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!
```

### 3. Index.php file code

```
ec2-user@ip-172-31-30-248:~/var/www/html/face
<?php
require_once(__DIR__ . '/vendor/autoload.php');

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

$bucket = 'aws-charan.s3.us-east-2.amazonaws.com';
$keyname = 'sample.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'
    ]);
}

// Print the URL to the object.
$imageUrl = $result['ObjectURL'];
if($imageUrl) {
    echo "Image upload done... Here is the URL: " . $imageUrl;
}

rekognition = new RekognitionClient([
    'region'      => 'us-east-2',
    'version'     => 'latest',
]);
$result = $rekognition->detectFaces([
    'Attributes'  => ['DEFAULT'],
    'Image'       => [
        'S3Object' => [
            'Bucket' => $bucket,
            'Name'  => $keyname,
            'Key'   => $keyname,
        ],
    ],
]);
echo "Totally there are " . count($result["FaceDetails"]) . " faces";
}
catch (Exception $e) {
    echo $e->getMessage() . PHP_EOL;
}

```

### 4. Upload Success Screenshot

```
ec2-user@ip-172-31-30-248:~/var/www/html/face
[ec2-user@ip-172-31-30-248 face]$ sudo php index.php
Image upload done... Here is the URL: https://s3.us-east-2.amazonaws.com/aws-charan.s30/sample.jpg
```

## Screenshots needed for EC2 and Rekognition

### 1. Face detect success screenshot

