

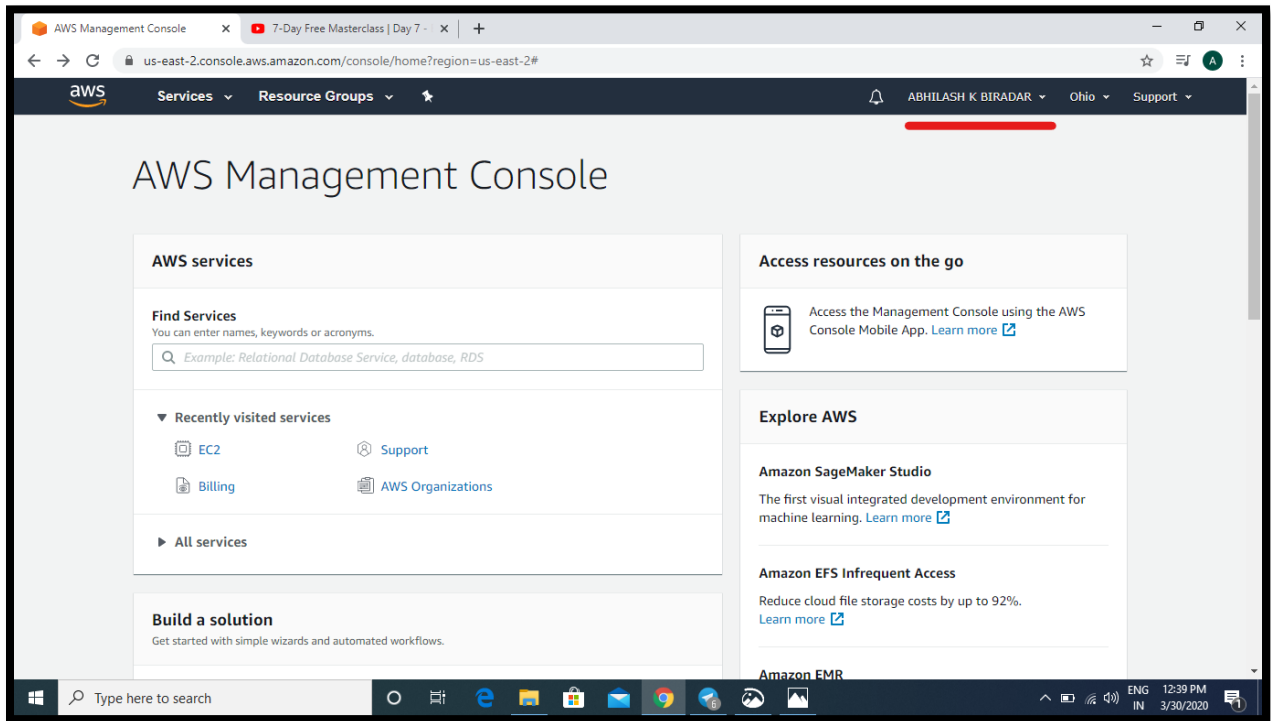


FACE-DETECTION USING AWS

Submitted by-

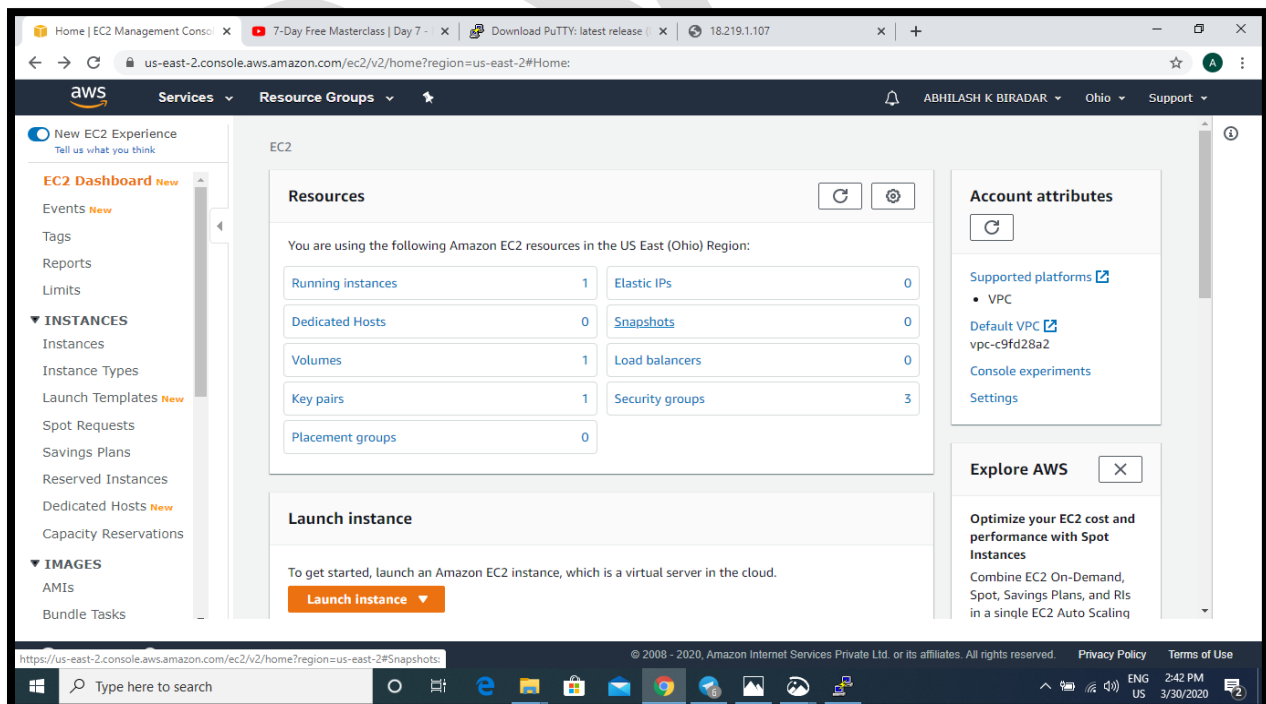
ABHILASH K BIRADAR

1. AWS LOGIN SCREEN WITH USERNAME

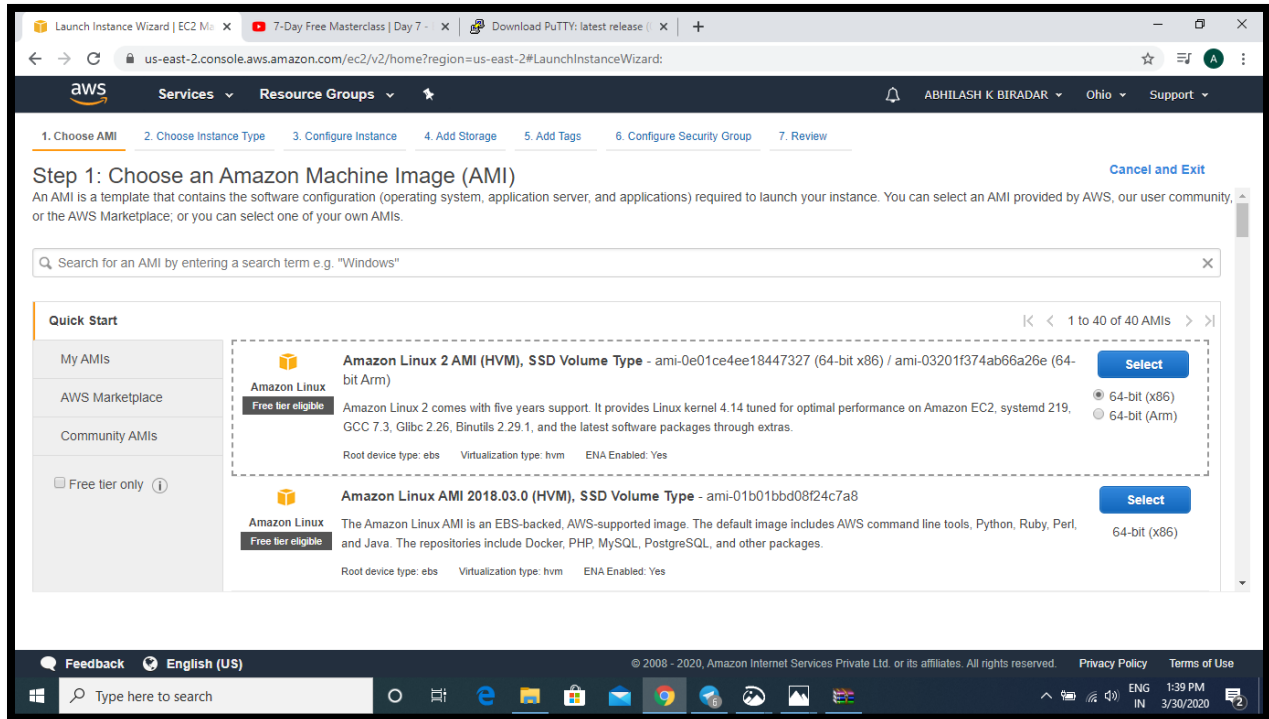


SCREEN SHOTS OF EC2

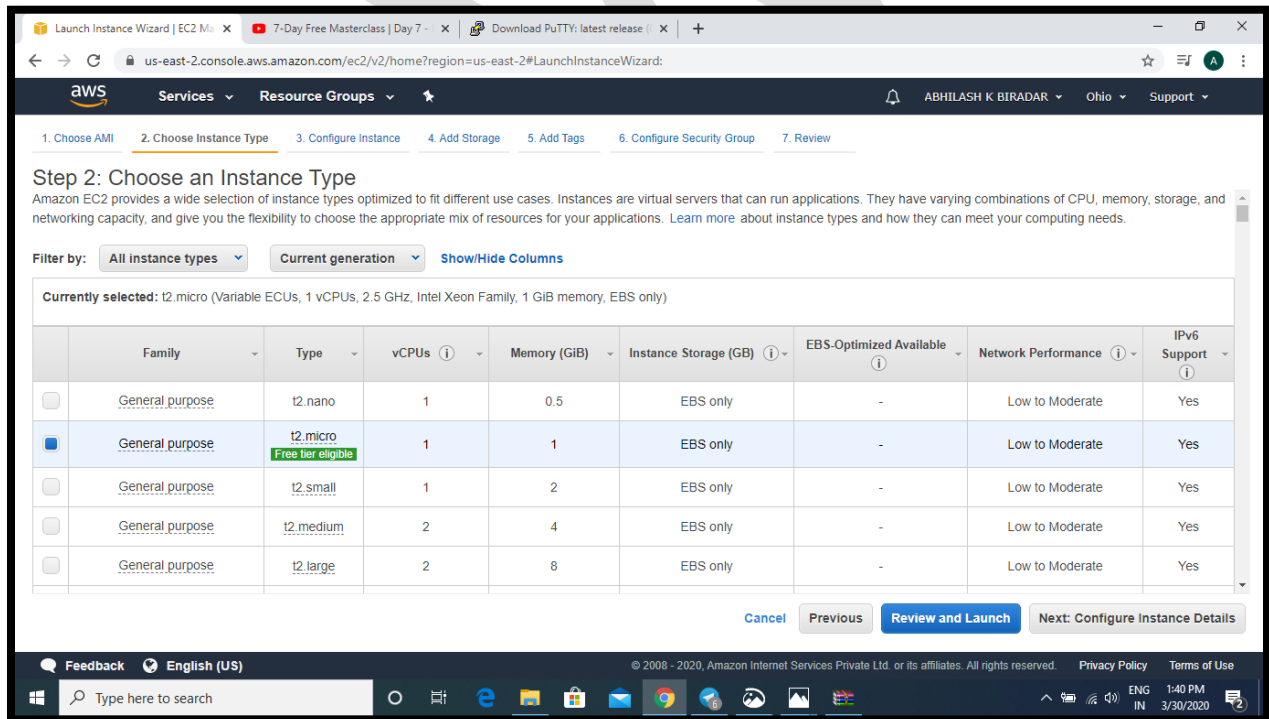
2. EC2 DASHBOARD



3. CHOOSING AN AMI



4. CHOOSING AN INSTANCE TYPE



5. ADDING STORAGE

Launch Instance Wizard | EC2 M5 | 7-Day Free Masterclass | Day 7 | Download PuTTY: latest release | +

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 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 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

6. CONFIGURING SECURITY GROUP

Launch Instance Wizard | EC2 M5 | 7-Day Free Masterclass | Day 7 | Download PuTTY: latest release | +

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

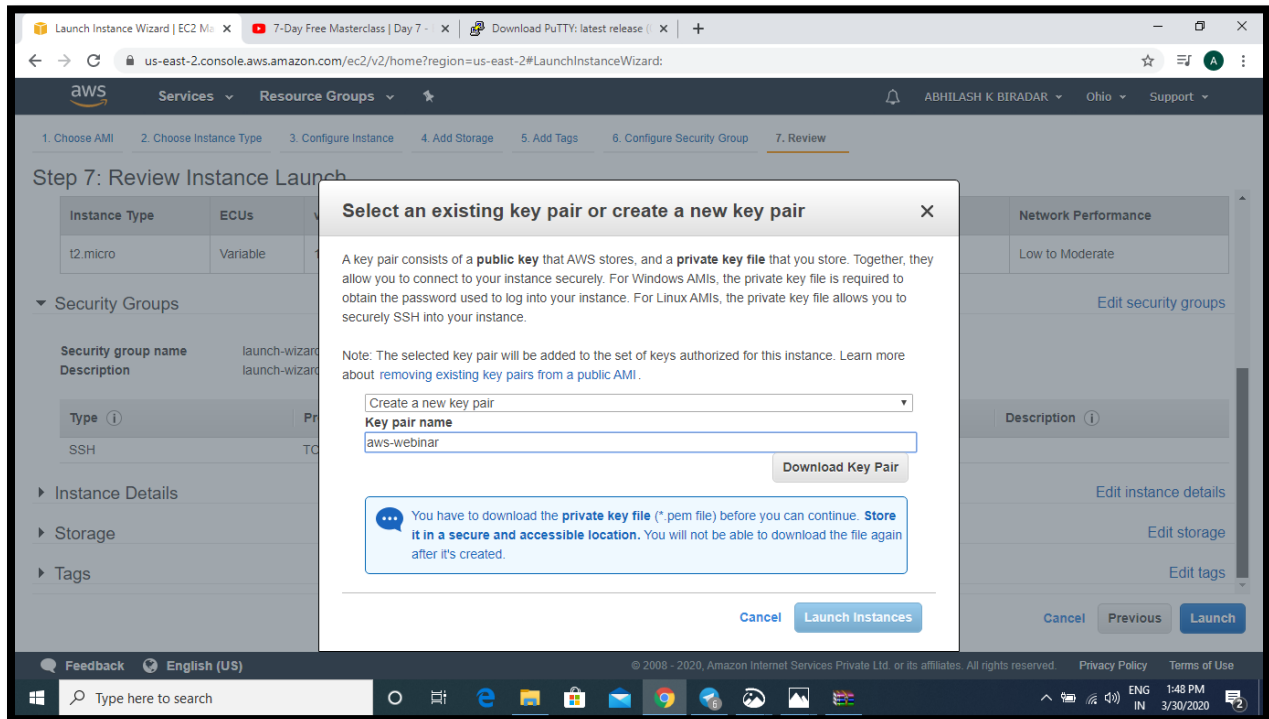
[Add Rule](#)

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.

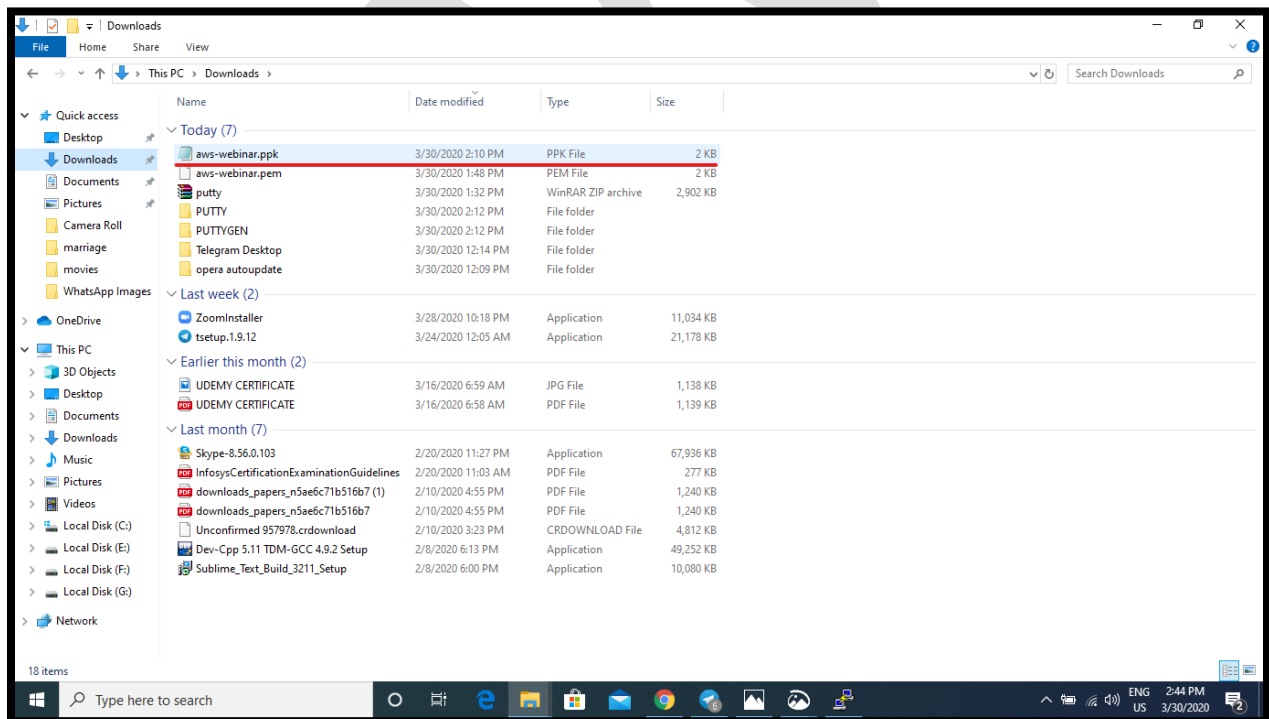
[Cancel](#) [Previous](#) [Review and Launch](#)

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

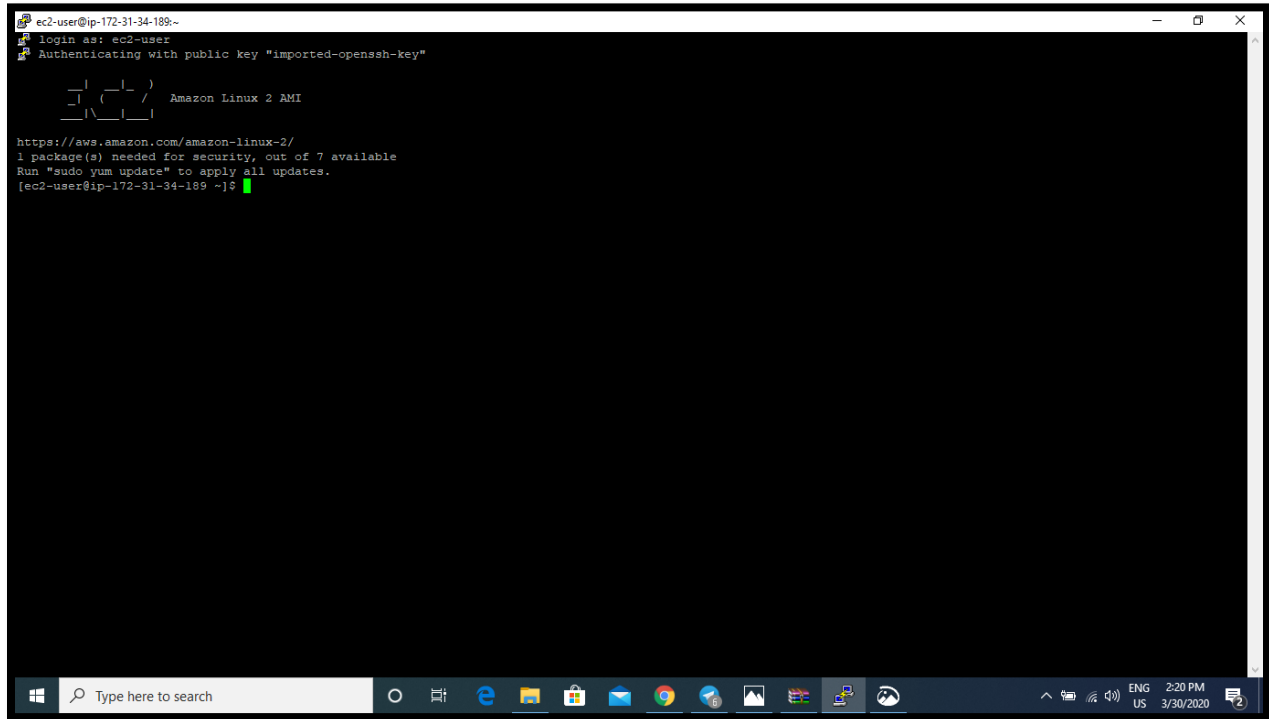
7. KEY PAIR DOWNLOAD



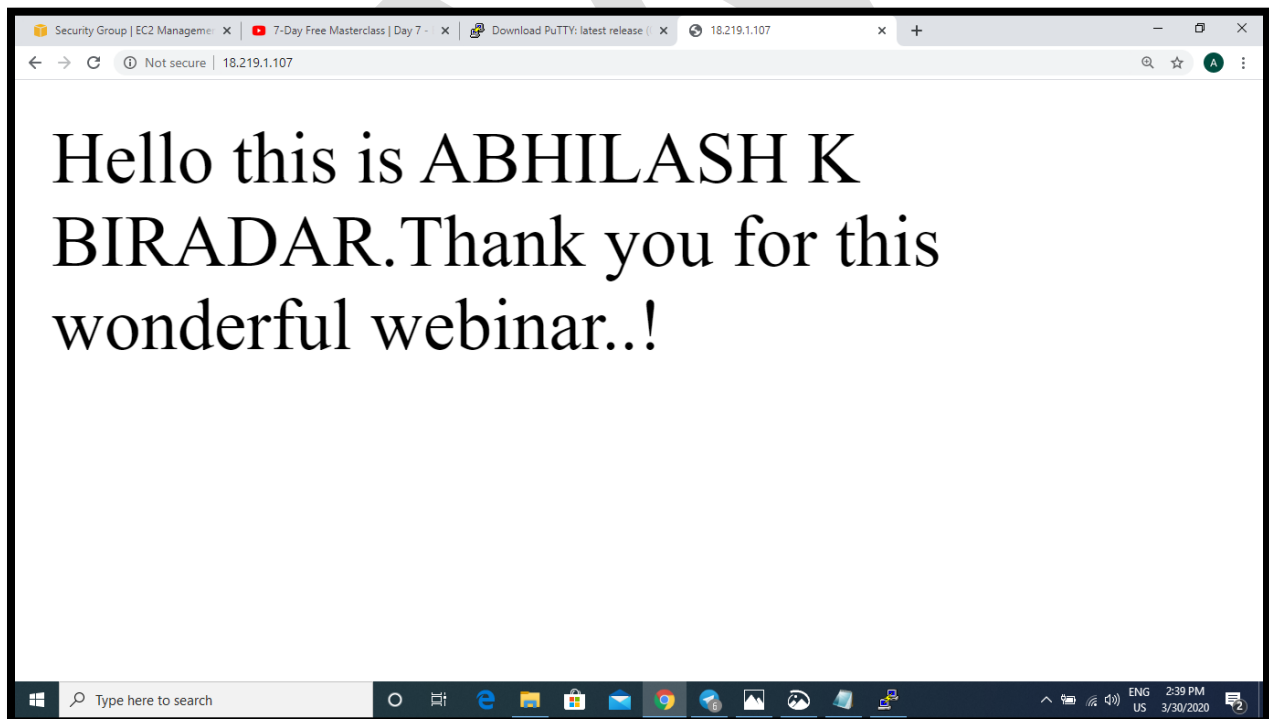
8. PuTTYGEN CONVERSION FILE FROM PEM TO PPK



9. LOGGED IN EC2 BLACKSCREEN

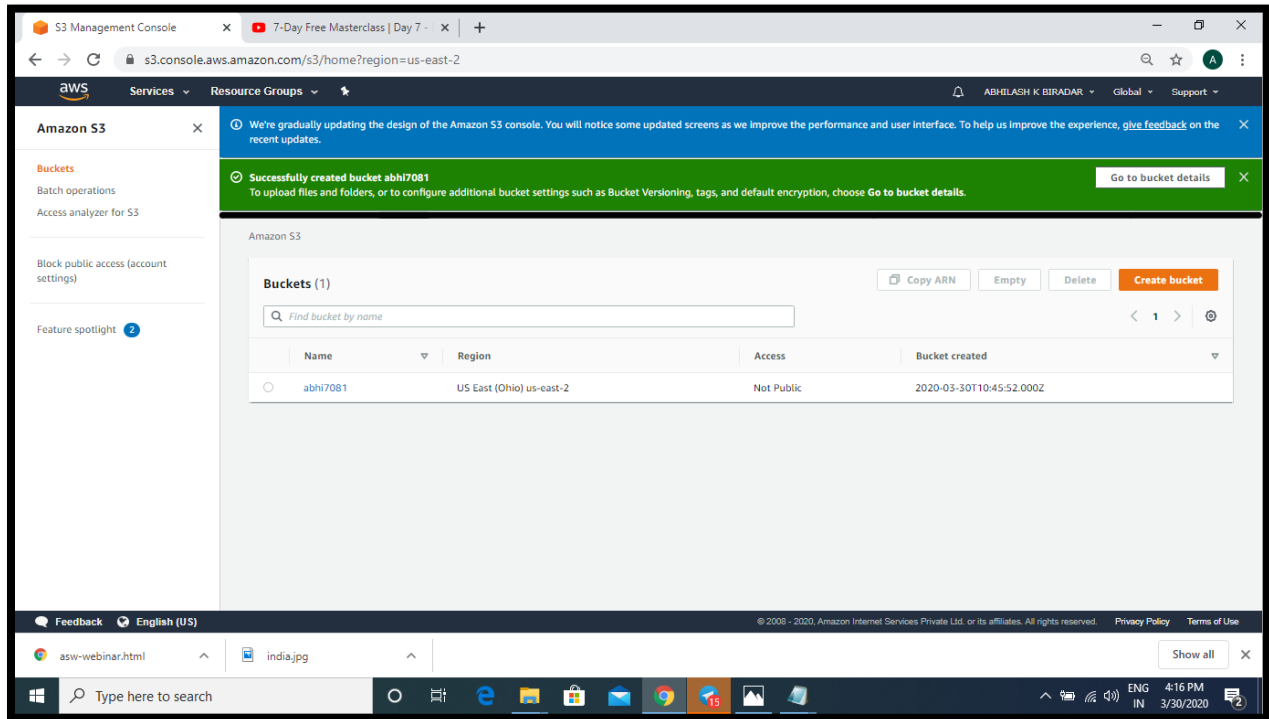


10. CONNECTED TO SERVER

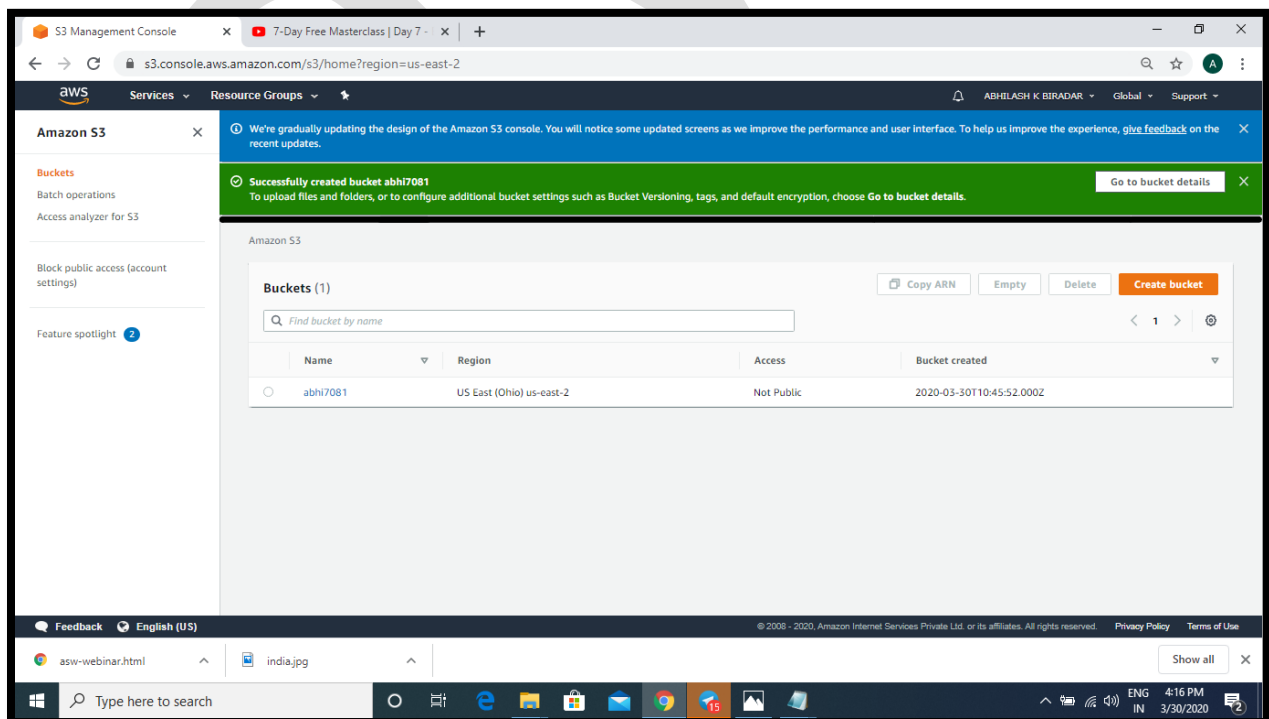


SCREEN SHOTS OF S3

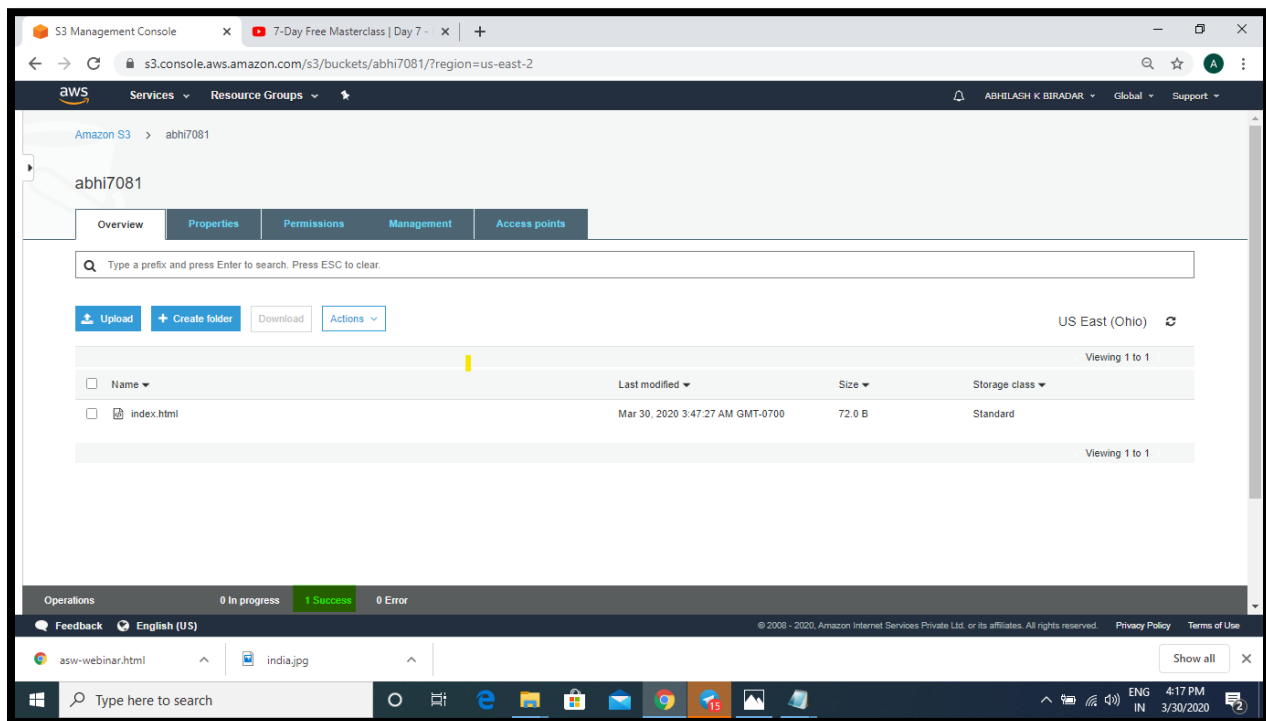
11. S3 DASHBOARD



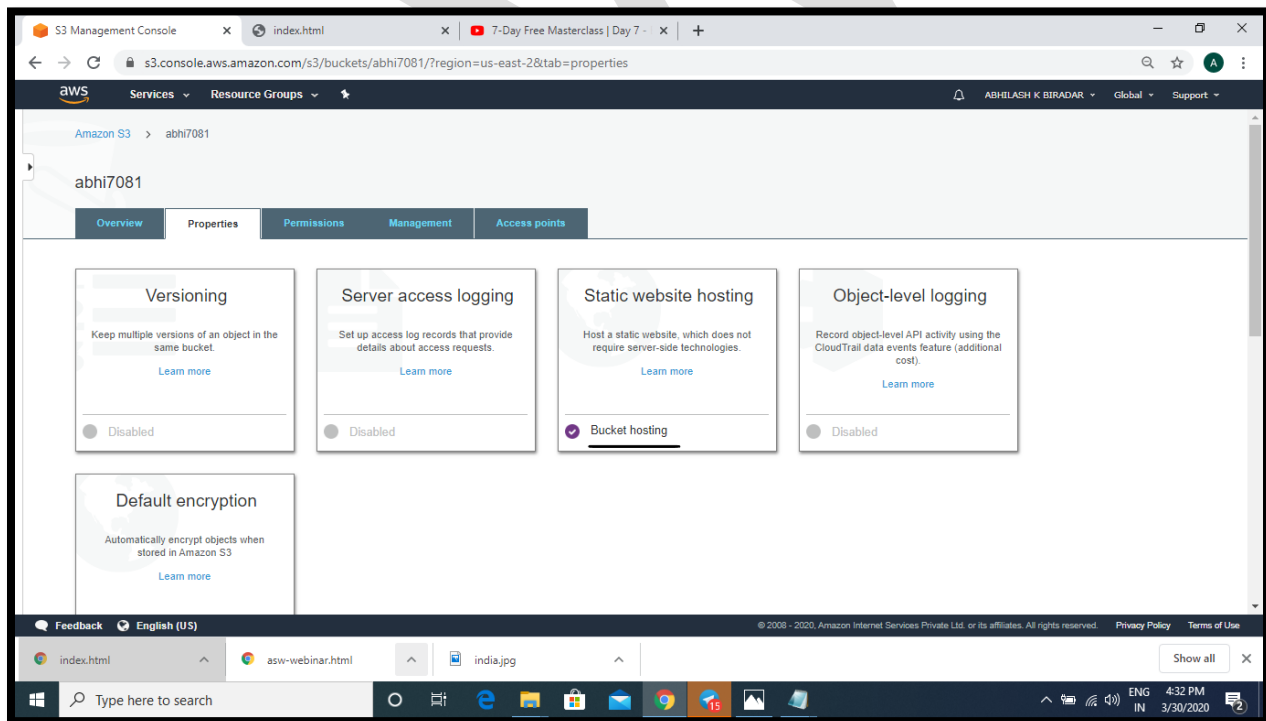
12. CREATING A BUCKET



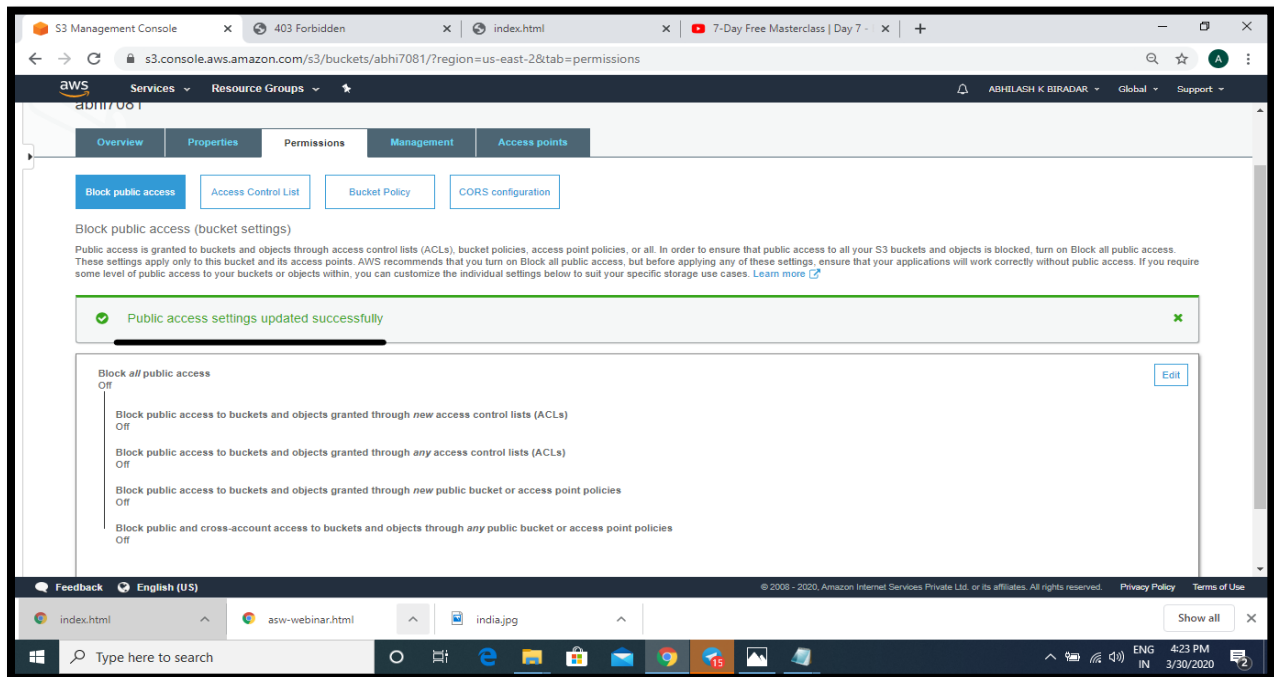
13. UPLOADING AN OBJECT



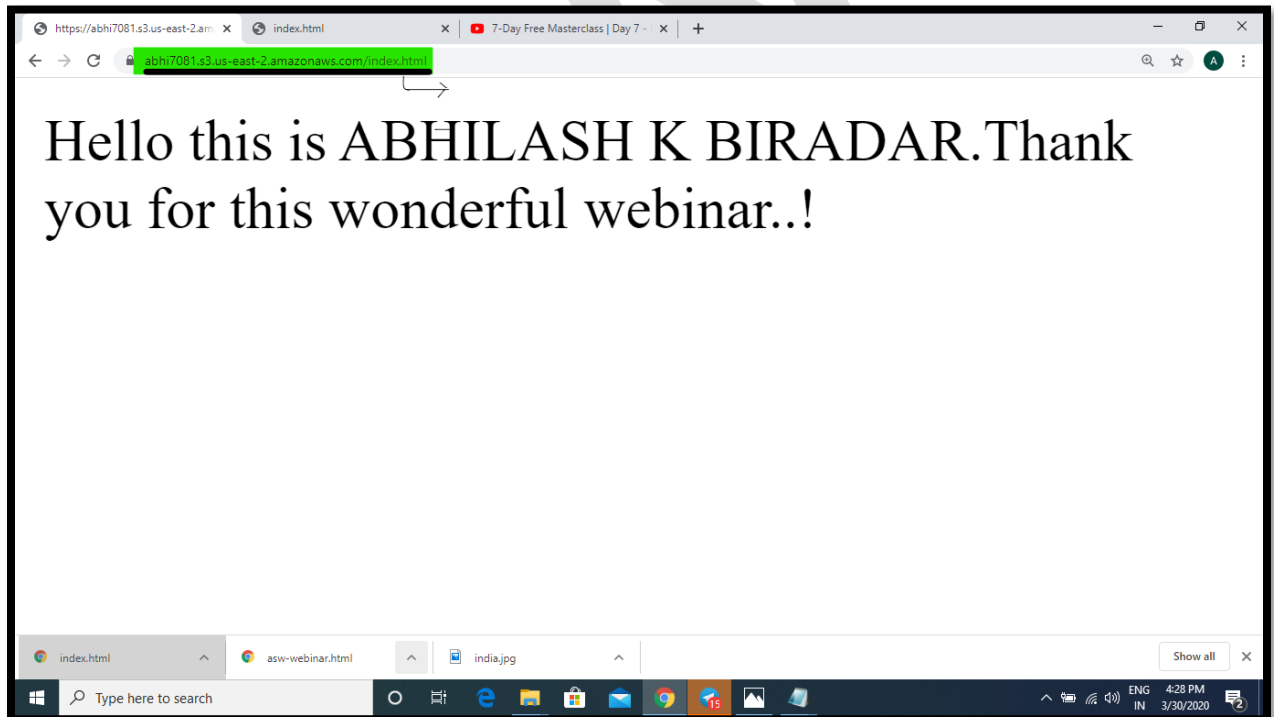
14. ENABLING A STATIC WEBSITE



15. MAKING THE OBJECT PUBLIC

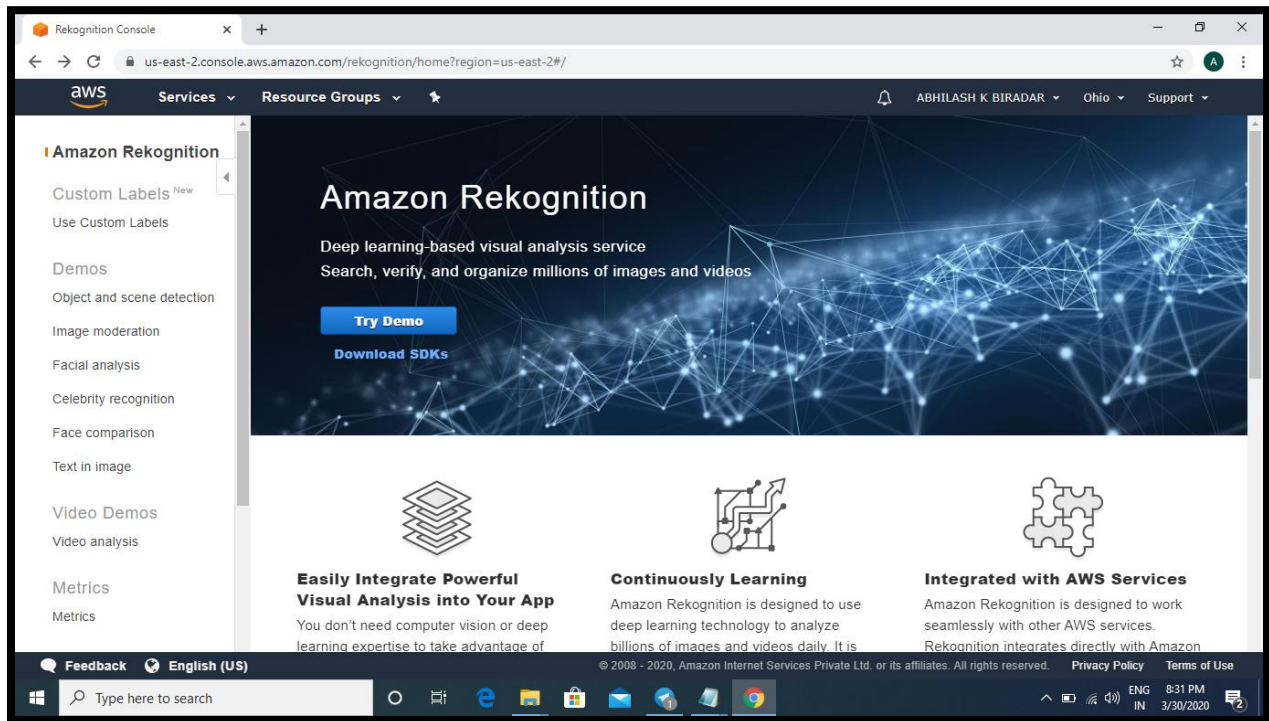


16. CHECKING THE S3 LINK ON THE BROWSER

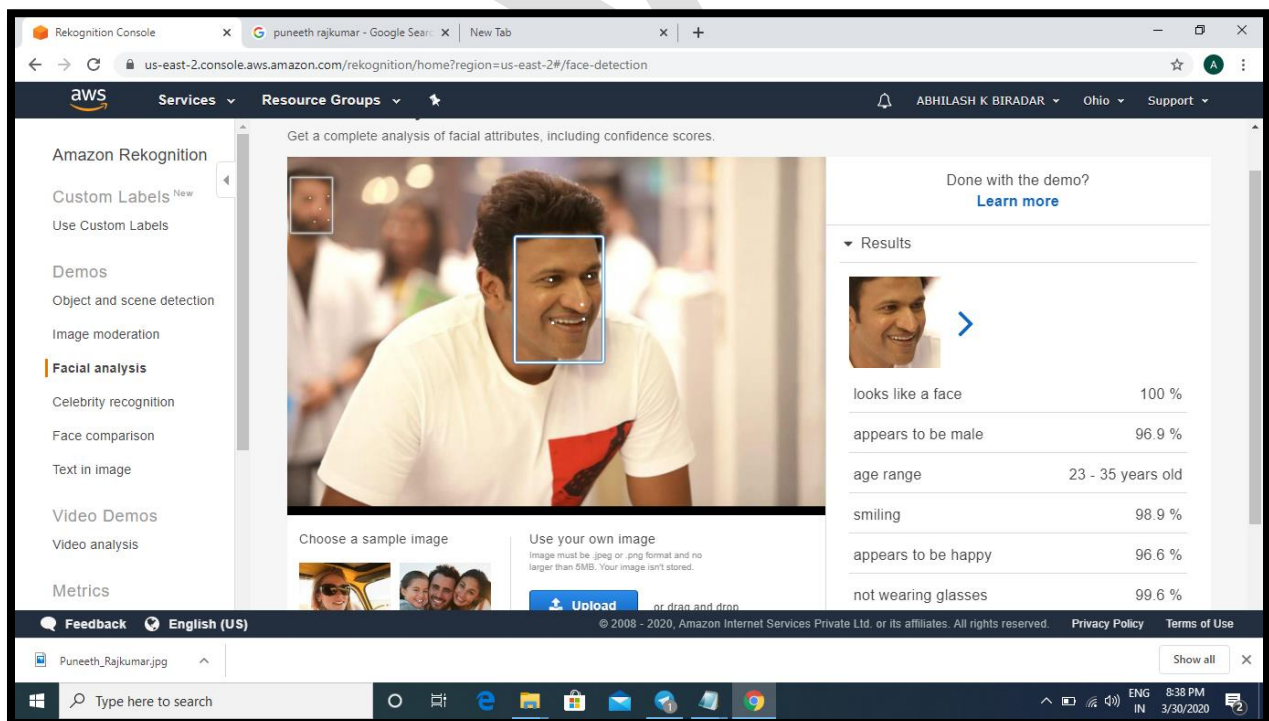


SCREEN SHOTS OF REKOGNITION

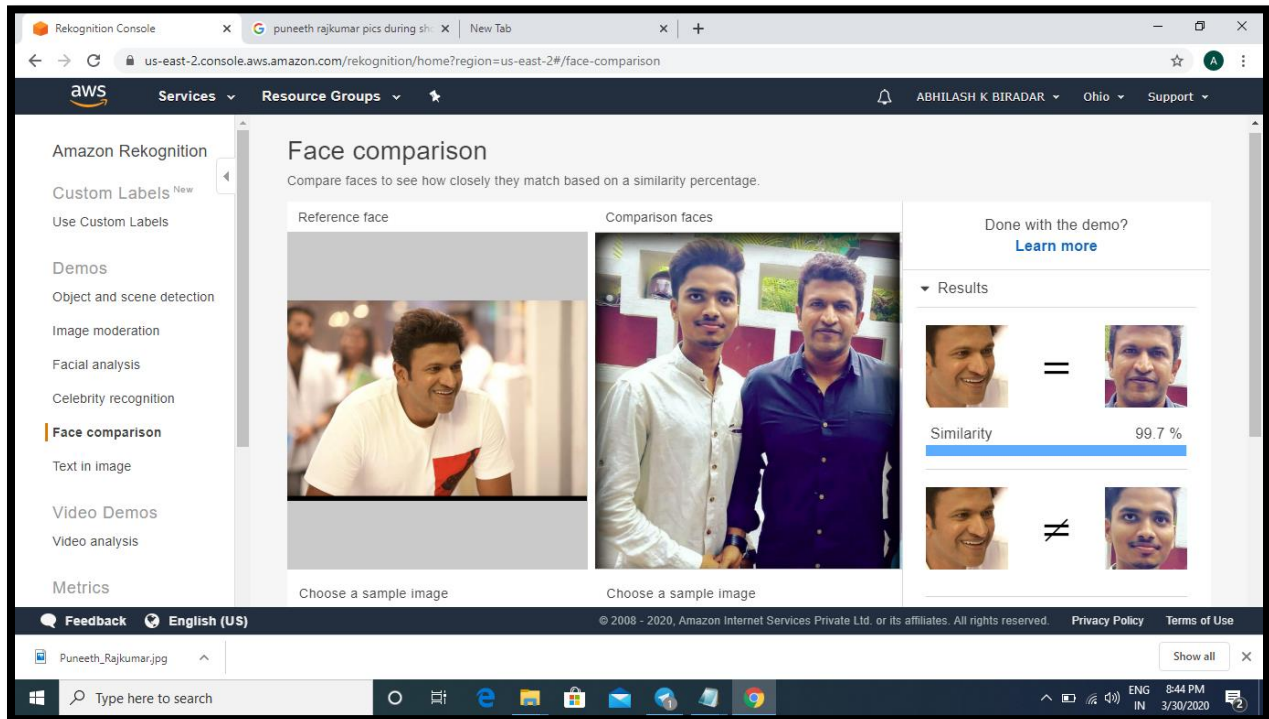
17. REKOGNITION DASHBOARD



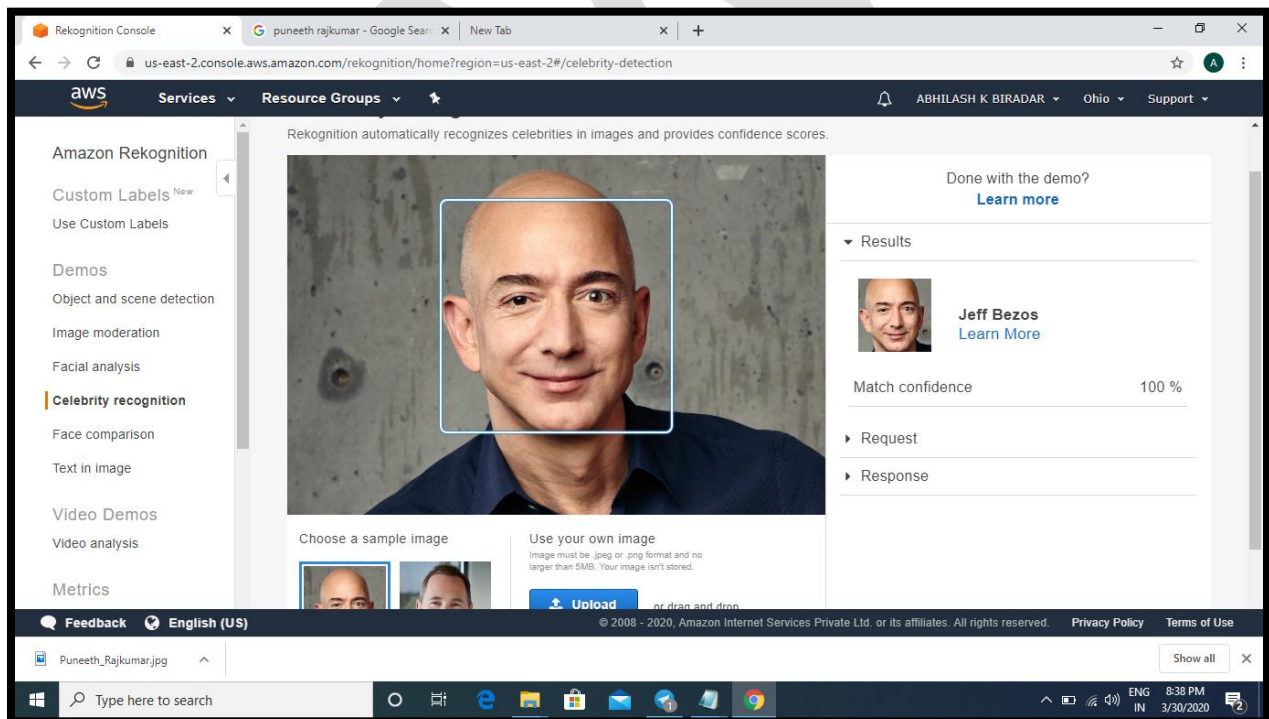
18. FACE DETECT



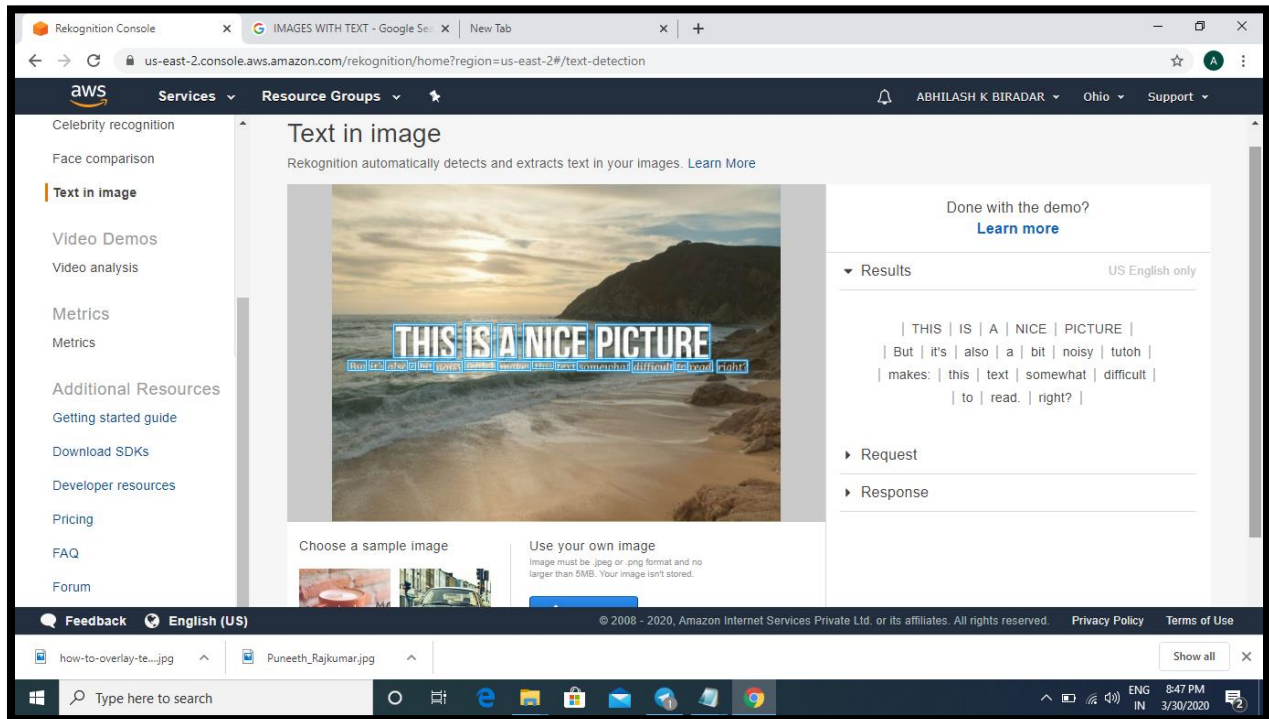
19. FACE COMPARE



20. CELEBRITY RECOGNITION



21. TEXT IN IMAGE



SCREEN SHOTS OF EC2 AND S3

22. INSTALLING AWS-SDK

```
ec2-user@ip-172-31-34-189:/var/www/html/face
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-172-31-34-189 ~]$ cd /var/www/html
[ec2-user@ip-172-31-34-189 html]$ sudo mkdir face
mkdir: cannot create directory 'face': File exists
[ec2-user@ip-172-31-34-189 html]$ cd face
[ec2-user@ip-172-31-34-189 face]$ pwd
/var/www/html/face
[ec2-user@ip-172-31-34-189 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
Generating autoload files
1 package you are using is looking for funding.
Use the `composer fund` command to find out more!
[ec2-user@ip-172-31-34-189 face]$
```

23. INSTALLING PHP

```
ec2-user@ip-172-31-34-189~$
Installing:
php      x86_64      5.4.16-46.amzn2.0.2      amzn2-core      1.4 M
Installing for dependencies:
libzip0- 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): libzip0-0.10.1-9.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 : libzip0-0.10.1-9.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  : libzip0-0.10.1-9.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:
  libzip0.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 ~]$
```


24. INDEX.PHP FILE CODE

```
ec2-user@ip-172-31-34-189:~$ sudo /bin/cd /dev/zero && /var/swap.1 bs"1M count=1024
sudo /bin/mkswap /var/swap.1
sudo /bin/swapon /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 = 'abhi7081';
$keyname = 'q.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 --
```

25. IMAGE IS SAVED

```
ec2-user@ip-172-31-34-189:~$ sudo wget https://images.newindianexpress.com/uploads/user/imagelibrary/2020/1/7/w1200X800/Puneeth_Rajkumar.jpg
--2020-03-30 15:31:23-- https://images.newindianexpress.com/uploads/user/imagelibrary/2020/1/7/w1200X800/Puneeth_Rajkumar.jpg
Resolving images.newindianexpress.com (images.newindianexpress.com)... 13.226.139.14, 13.226.139.45, 13.226.139.110, ...
Connecting to images.newindianexpress.com (images.newindianexpress.com)|13.226.139.14|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 373249 (365K) [image/jpeg]
Saving to: 'Puneeth_Rajkumar.jpg.1'

100%[=====] 373,249 --.-K/s in 0.07s

2020-03-30 15:31:23 (5.23 MB/s) - 'Puneeth_Rajkumar.jpg.1' saved [373249/373249]

[ec2-user@ip-172-31-34-189 face]$
```

26. IMAGE UPLOAD SUCCESS

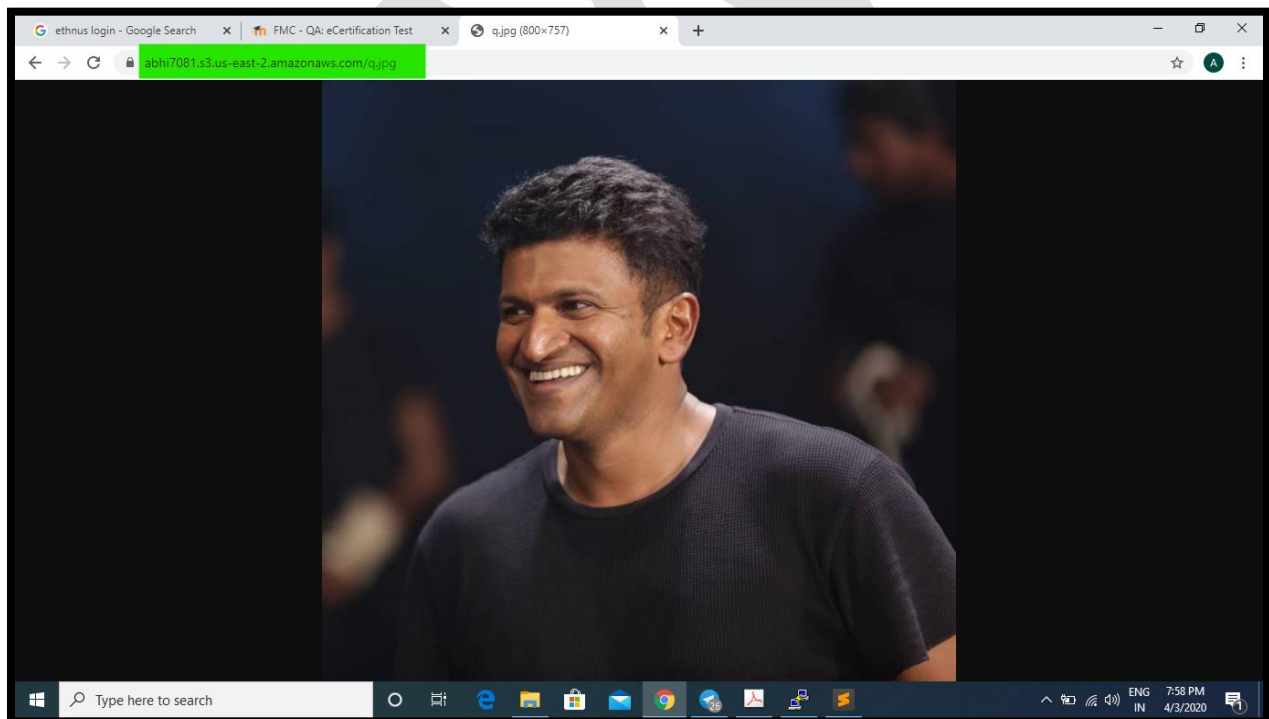
```
ec2-user@ip-172-31-34-189:/var/www/html/face
[ec2-user@ip-172-31-34-189 ~]$ sudo yum install httpd
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core | 2.4 kB | 00:00
amzn2extra-docker | 1.8 kB | 00:00
amzn2extra-php7.2 | 1.8 kB | 00:00
Package httpd-2.4.41-1.amzn2.0.1.x86_64 already installed and latest version
Nothing to do
[ec2-user@ip-172-31-34-189 ~]$ sudo yum install php
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Package php-7.2.28-1.amzn2.x86_64 already installed and latest version
Nothing to do
[ec2-user@ip-172-31-34-189 ~]$ 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-172-31-34-189 ~]$ cd /var/www/html
[ec2-user@ip-172-31-34-189 html]$ sudo mkdir face
mkdir: cannot create directory 'face': File exists
[ec2-user@ip-172-31-34-189 html]$ cd face
[ec2-user@ip-172-31-34-189 face]$ pwd
/var/www/html/face
[ec2-user@ip-172-31-34-189 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)
Package operations: 0 installs, 1 update, 0 removals
  - Updating aws/aws-sdk-php (3.133.47 => 3.134.2): Downloading (100%)
Waiting lock file
Generating autoload files
1 package you are using is looking for funding.
Use the 'composer fund' command to find out more!
[ec2-user@ip-172-31-34-189 face]$
[ec2-user@ip-172-31-34-189 face]$ sudo vim index.php
[ec2-user@ip-172-31-34-189 face]$ sudo php index.php
Image upload done... Here is the URL: https://abhi7081.s3.us-east-2.amazonaws.com
[ec2-user@ip-172-31-34-189 face]$
```

27. UPLOADED IMAGE OPENED IN BROWSER USING URL



SCREENSHOTS OF EC2 AND REKOGNITION

28. PHP CODE FOR REKOGNITION

```
ec2-user@ip-172-31-34-189:~
require_once(__DIR__ . '/vendor/autoload.php');

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

$bucket = 'abhi7081';
$keyname = 'q.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,
                ],
            ],
        ]);
    }
} catch (Exception $e) {
    echo $e->getMessage();
}

-- INSERT --
```

29. FACE DETECT SUCCESS

```
ec2-user@ip-172-31-34-189:/var/www/html/face
[ec2-user@ip-172-31-34-189 ~]$ cd /var/www/html/face
[ec2-user@ip-172-31-34-189 face]$ pwd
/var/www/html/face
[ec2-user@ip-172-31-34-189 face]$ ls
b97ea33b5842c7894b804923c6c05580.jpg.1  PRK.jpg
composer.json                          puneeth-rajkumar_155247890030.jpg.1
composer.lock                          Puneeth_Rajkumar.jpg.1
index.php                              q.jpg
P.jpg                                  sample.jpg
prk.jpg                                vendor
[ec2-user@ip-172-31-34-189 face]$ sudo vim index.php
[ec2-user@ip-172-31-34-189 face]$ sudo vim index.php
[ec2-user@ip-172-31-34-189 face]$ sudo php index.php
Image upload done... Here is the URL: https://abhi7081.s3.us-east-2.amazonaws.com/sample.jpgTotally there are 9 faces[ec2-user@ip-172-31-34-189 face]$ sudo php index.php
Image upload done... Here is the URL: https://abhi7081.s3.us-east-2.amazonaws.com/sample.jpgTotally there are 9 faces[ec2-user@ip-172-31-34-189 face]$ sudo vim index.php
[ec2-user@ip-172-31-34-189 face]$ sudo php index.php
Image upload done... Here is the URL: https://abhi7081.s3.us-east-2.amazonaws.com/sample.jpgTotally there are 9 faces[ec2-user@ip-172-31-34-189 face]$ sudo php index.php
Image upload done... Here is the URL: https://abhi7081.s3.us-east-2.amazonaws.com/sample.jpgTotally there are 9 faces[ec2-user@ip-172-31-34-189 face]$ sudo php index.php
Image upload done... Here is the URL: https://abhi7081.s3.us-east-2.amazonaws.com/q.jpgTotally there are 1 faces[ec2-user@ip-172-31-34-189 face]$
```