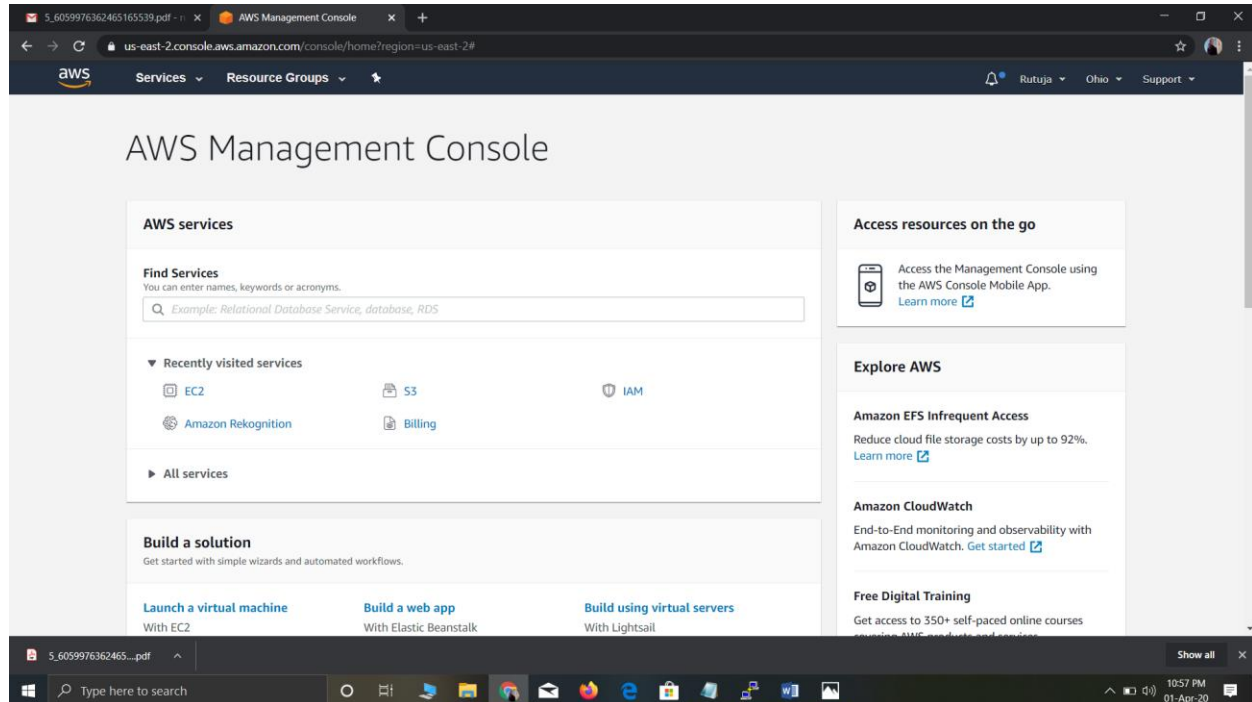
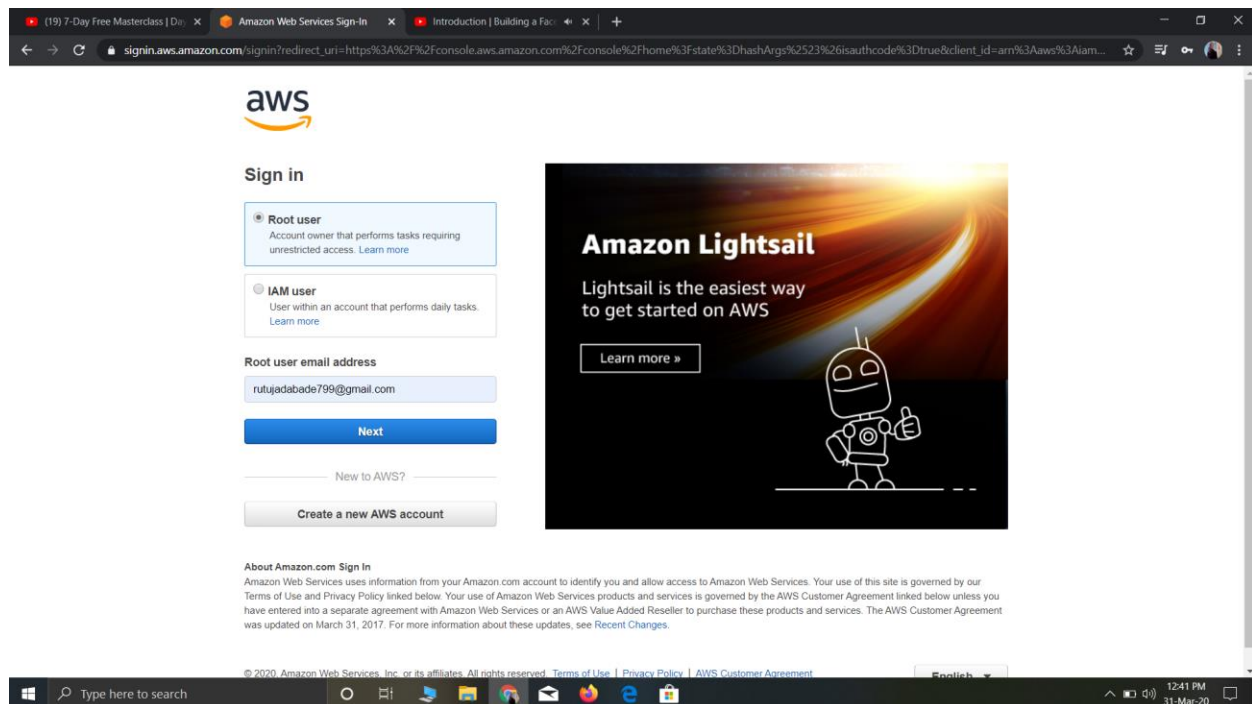


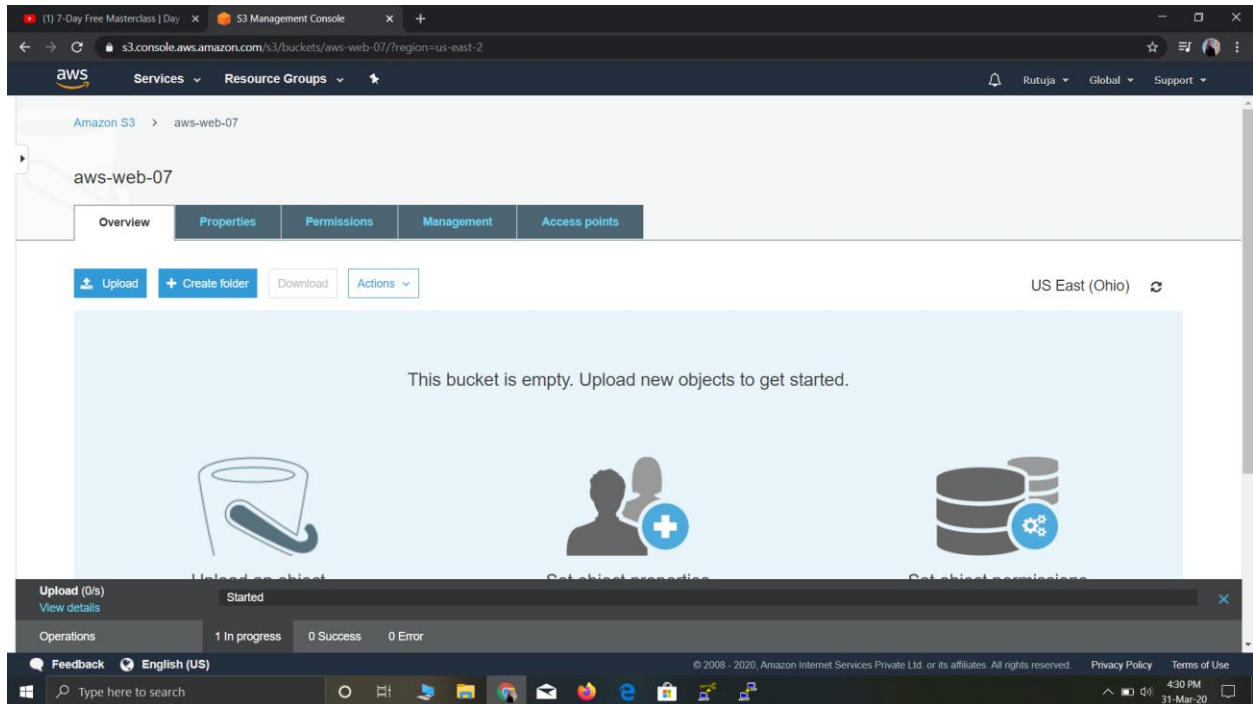
1. AWS Login screen with username



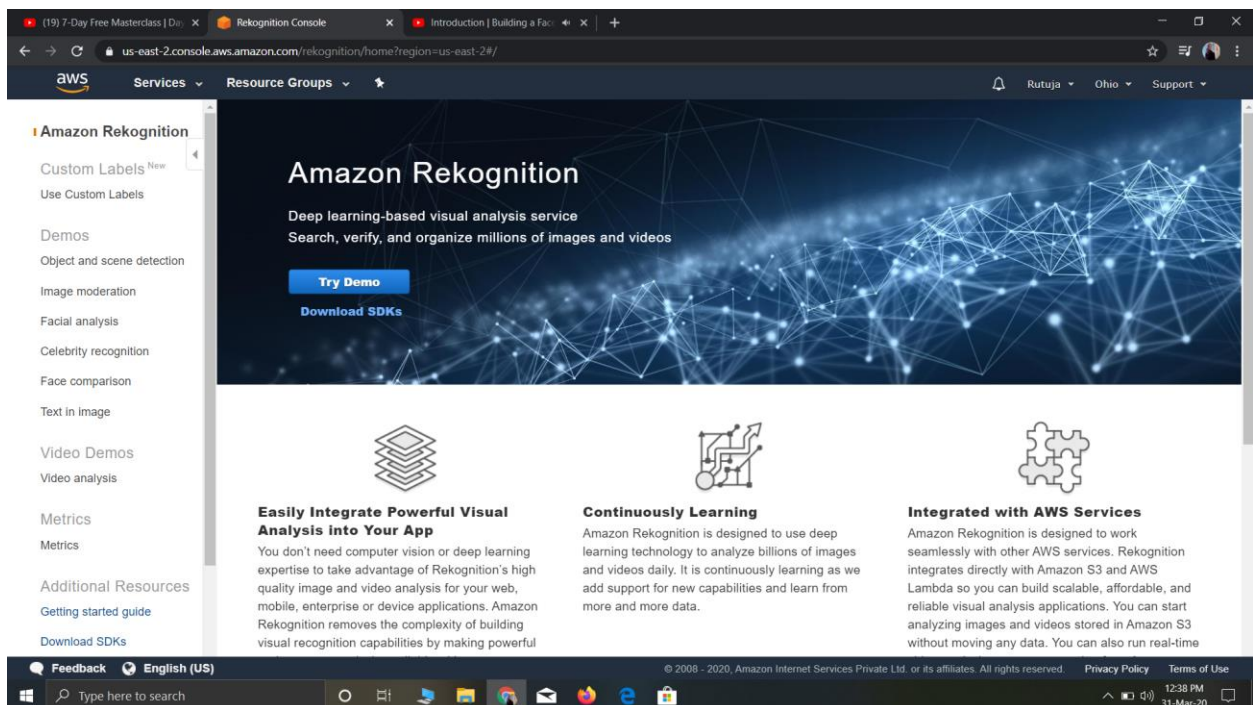
2. EC2 Dashboard



3.S3 Dashboard



4.Rekognition Dashboard



Screenshots needed for EC2

1. Choosing an AMI

The screenshot shows the AWS Management Console's 'Launch instance wizard' for EC2. The 'Choose AMI' step is active, displaying a search bar and a list of AMIs. The 'Quick Start' section on the left shows 'My AMIs', 'AWS Marketplace', and 'Community AMIs'. The main list shows three AMIs: Amazon Linux 2 AMI (HVM, SSD Volume Type), Amazon Linux AMI 2018.03.0 (HVM, SSD Volume Type), and Red Hat Enterprise Linux 8 (HVM, SSD Volume Type). Each AMI has a 'Select' button and radio buttons for 64-bit (x86) and 64-bit (Arm).

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 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 ENA Enabled: Yes

Amazon Linux AMI 2018.03.0 (HVM, SSD Volume Type - ami-01b01bbd08f24c7a8)

Amazon Linux Free tier eligible

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 Free tier eligible

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

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

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

2. Choosing an Instance Type

The screenshot shows the AWS Management Console's 'Launch instance wizard' for EC2. The 'Choose Instance Type' step is active, displaying a table of instance types. The 'Filter by' section shows 'All instance types', 'Current generation', and 'Show/Hide Columns'. The table lists various instance types, including t2.nano, t2.micro, t2.small, t2.medium, t2.large, t2.xlarge, t2.2xlarge, and t3a.nano. The 't2.micro' instance type is highlighted as 'Free tier eligible'.

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	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
<input type="checkbox"/>	General purpose	t3a.nano	2	0.5	EBS only	Yes	Up to 5 Gigabit	Yes

Cancel Previous Review and Launch Next: Configure Instance Details

© 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 Management Console's 'Launch instance wizard' for an EC2 instance. The '4. Add Storage' step is active. It displays a table for the root volume configuration.

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

Below the table is an 'Add New Volume' button. A blue information box 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 navigation buttons: 'Cancel', 'Previous', 'Review and Launch' (highlighted), and 'Next: Add Tags'.

4.Configuring Security Group

The screenshot shows the AWS Management Console's 'Launch instance wizard' for an EC2 instance. The '6. Configure Security Group' step is active. It shows the configuration for a new security group.

Assign a security group: ☒ Create a new security group ☐ Select an existing security group

Security group name:

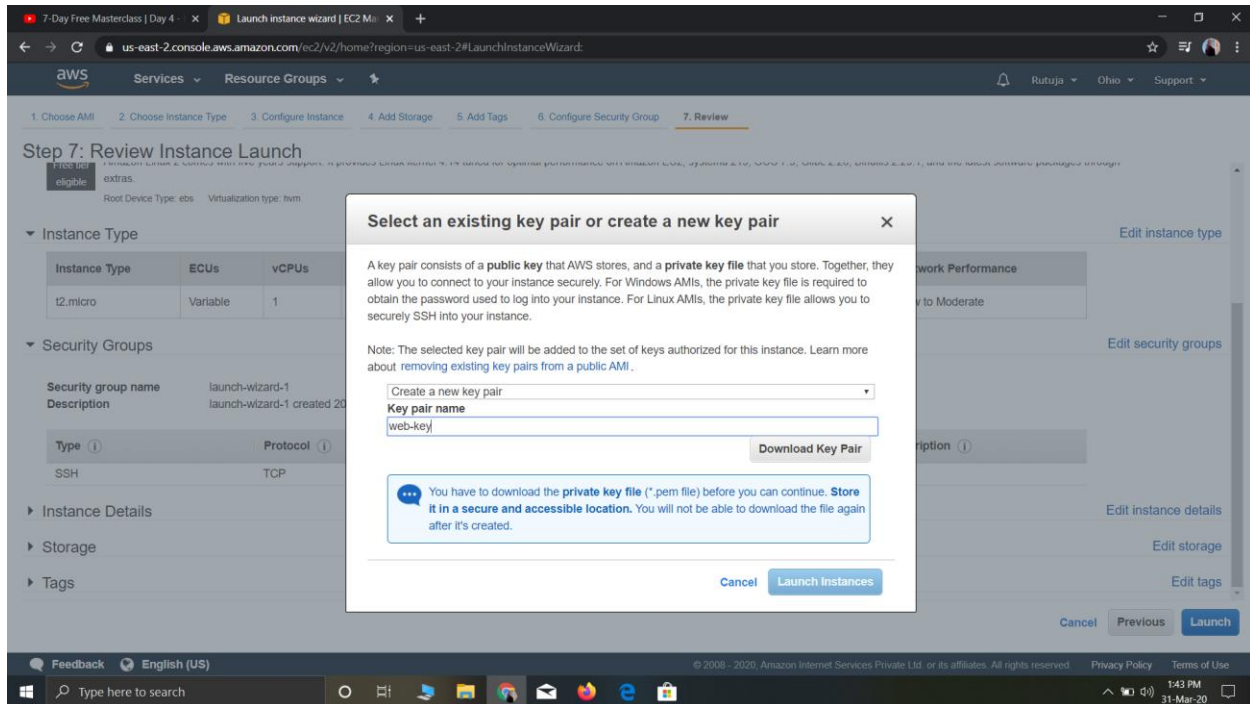
Description:

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop

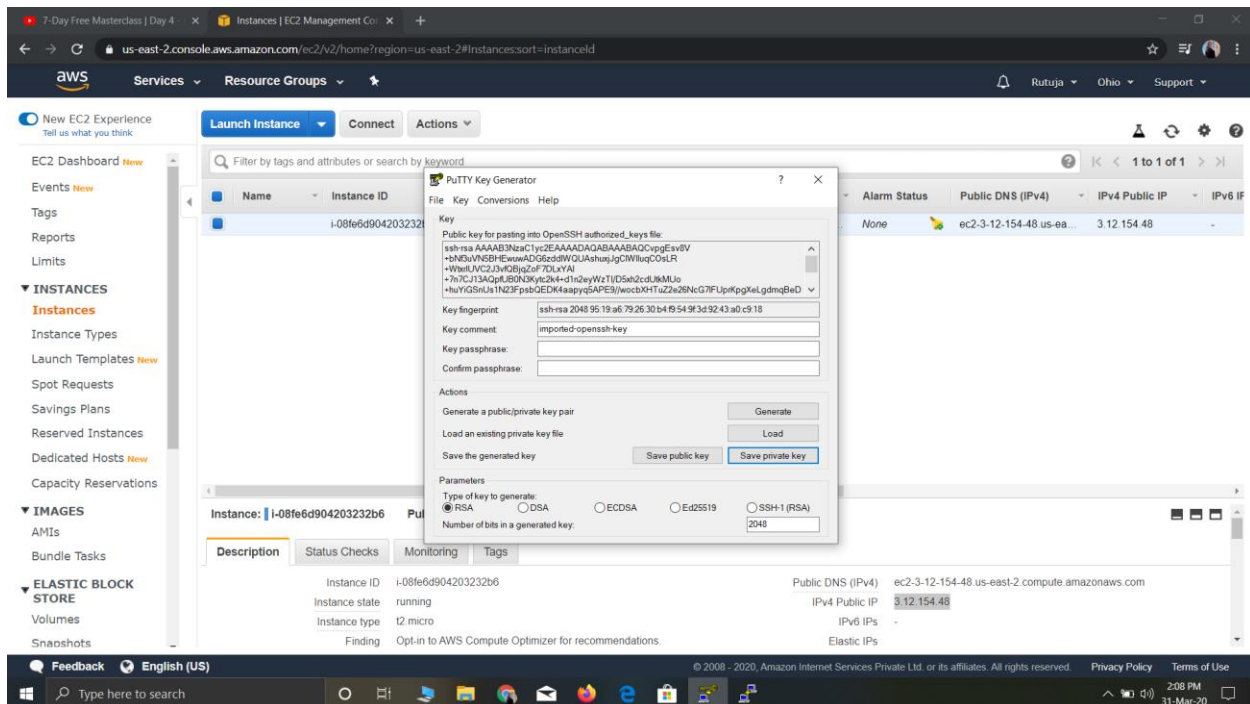
Below the table is an 'Add Rule' button. A yellow warning box states: 'Warning: Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.'

At the bottom right, there are navigation buttons: 'Cancel', 'Previous', 'Review and Launch' (highlighted), and 'Next: Add Tags'.

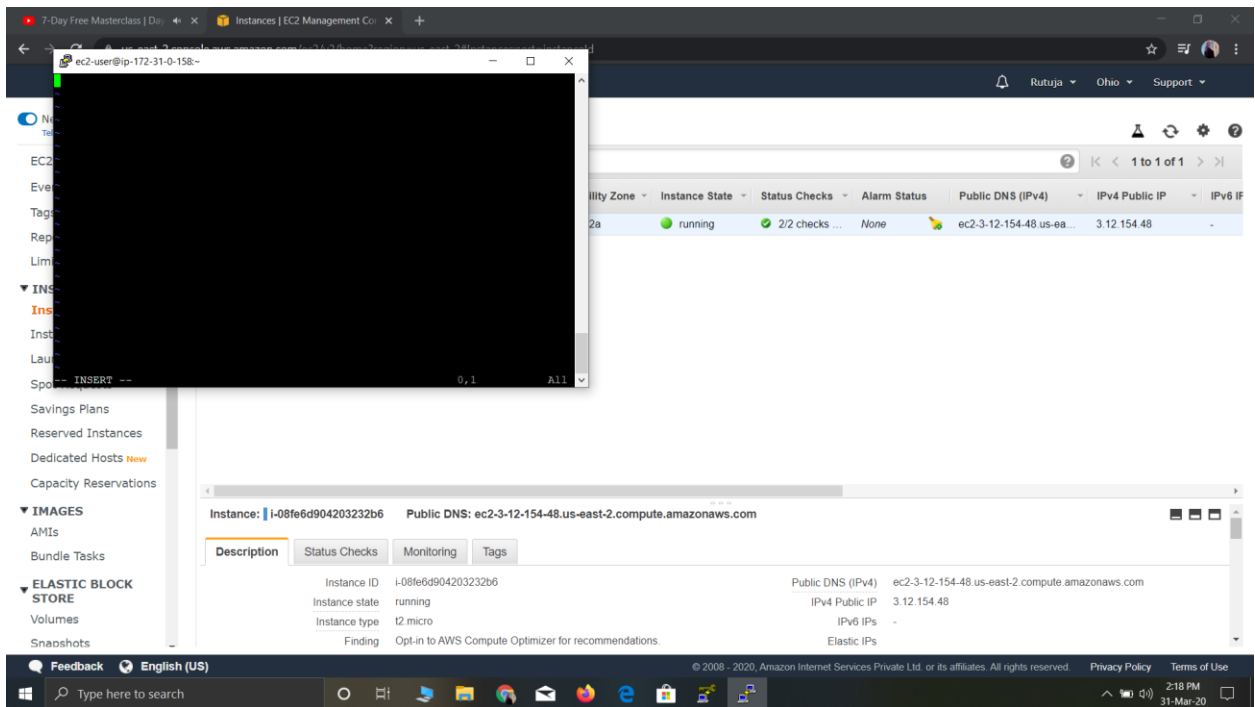
5. Key Pair Download



6. PuTTYgen conversion from pem to ppk

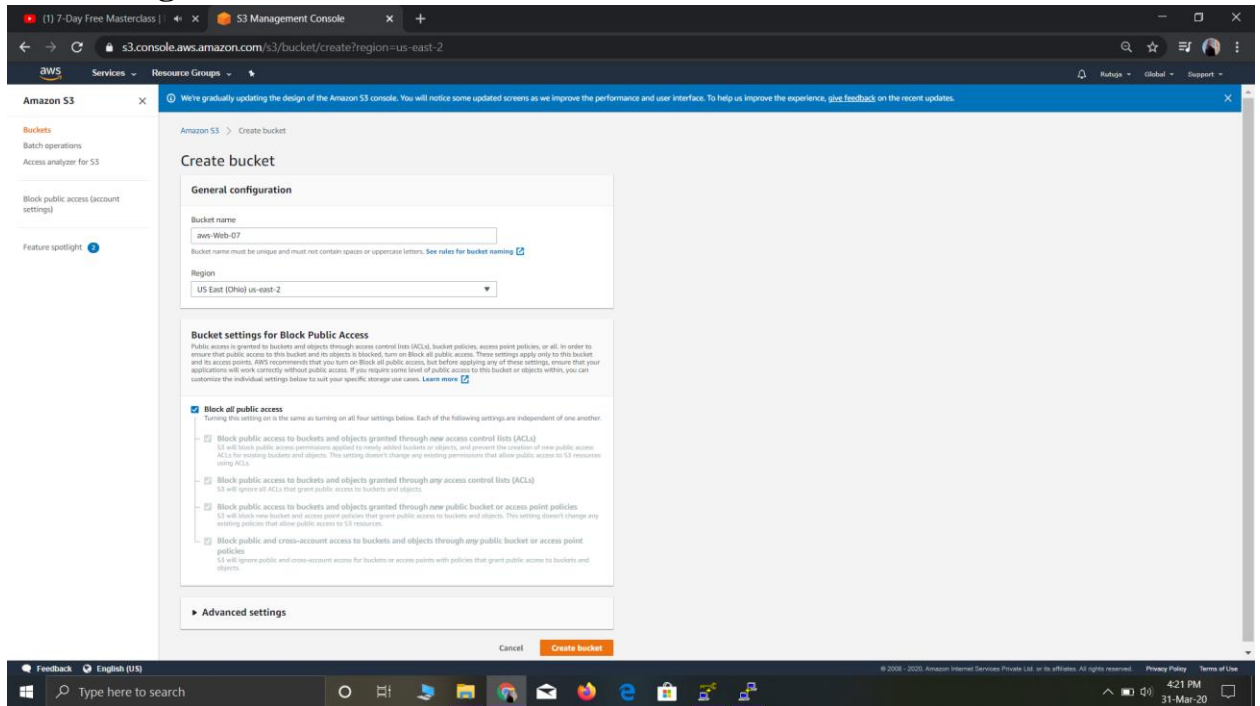


7. Logged in EC2 black screen

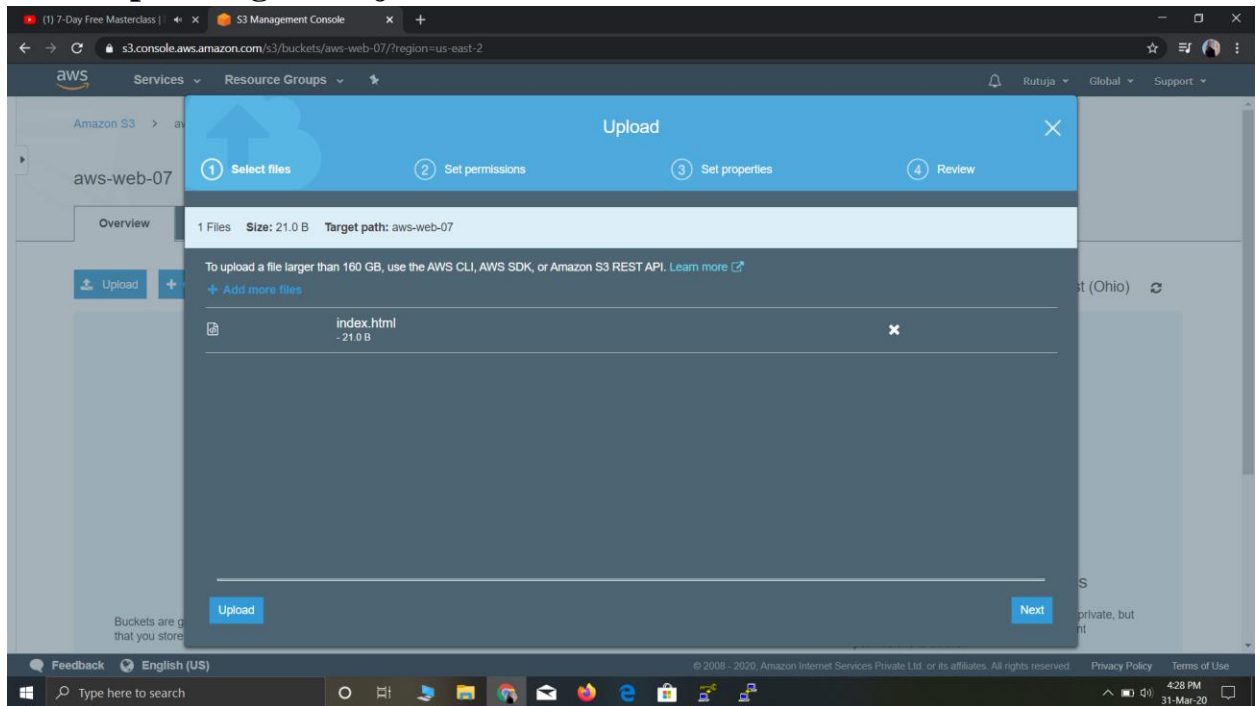


Screenshots needed for S3

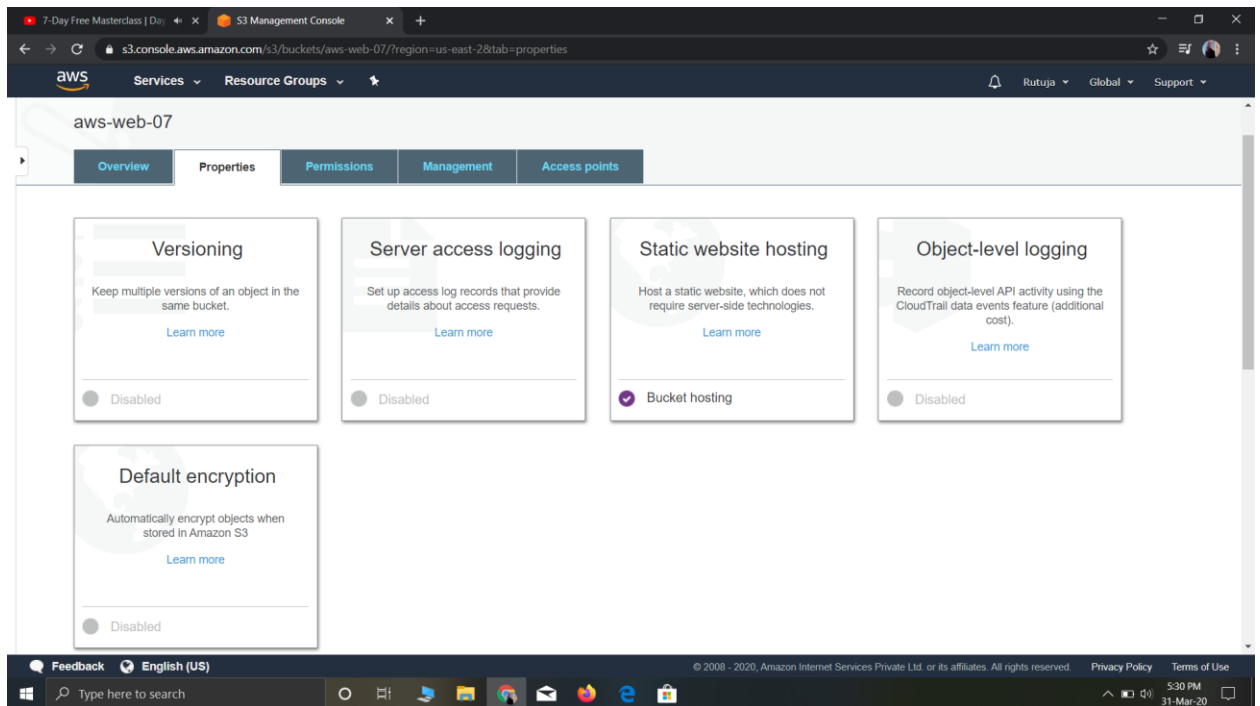
1. Creating a bucket



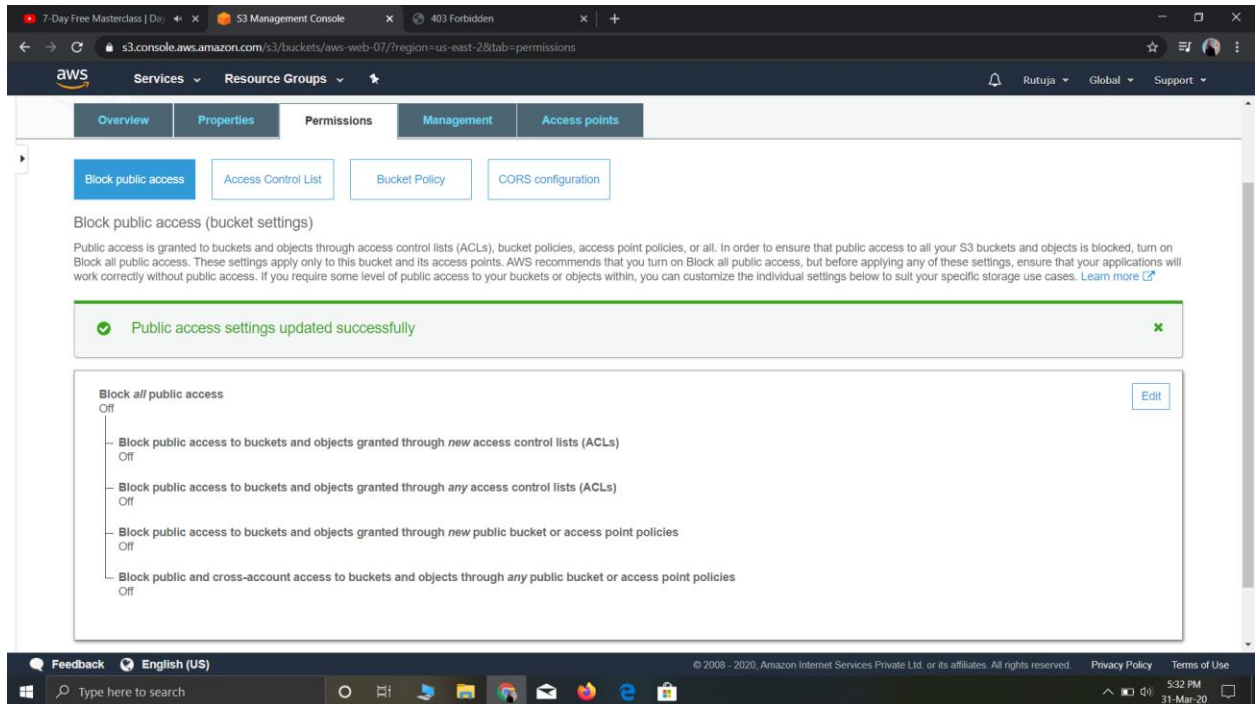
2. Uploading an Object



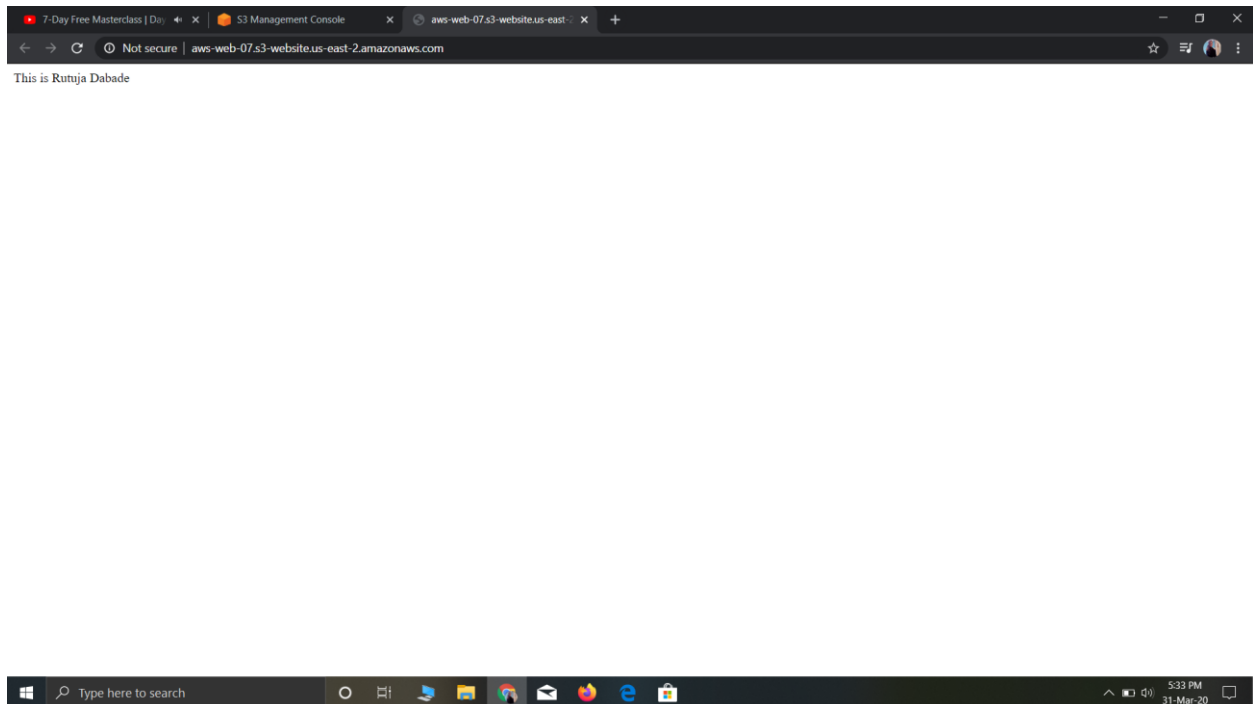
3. Enabling Static Website



4. Making the Object Public

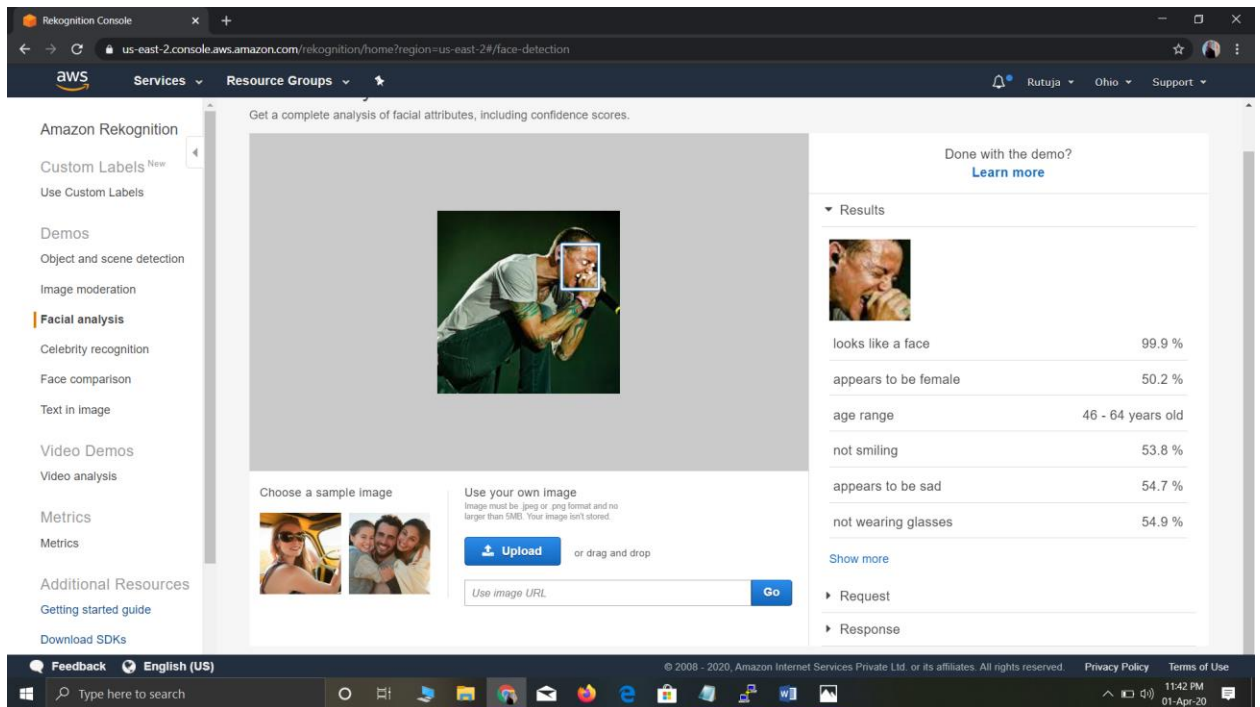


5. Checking the S3 link on the browser

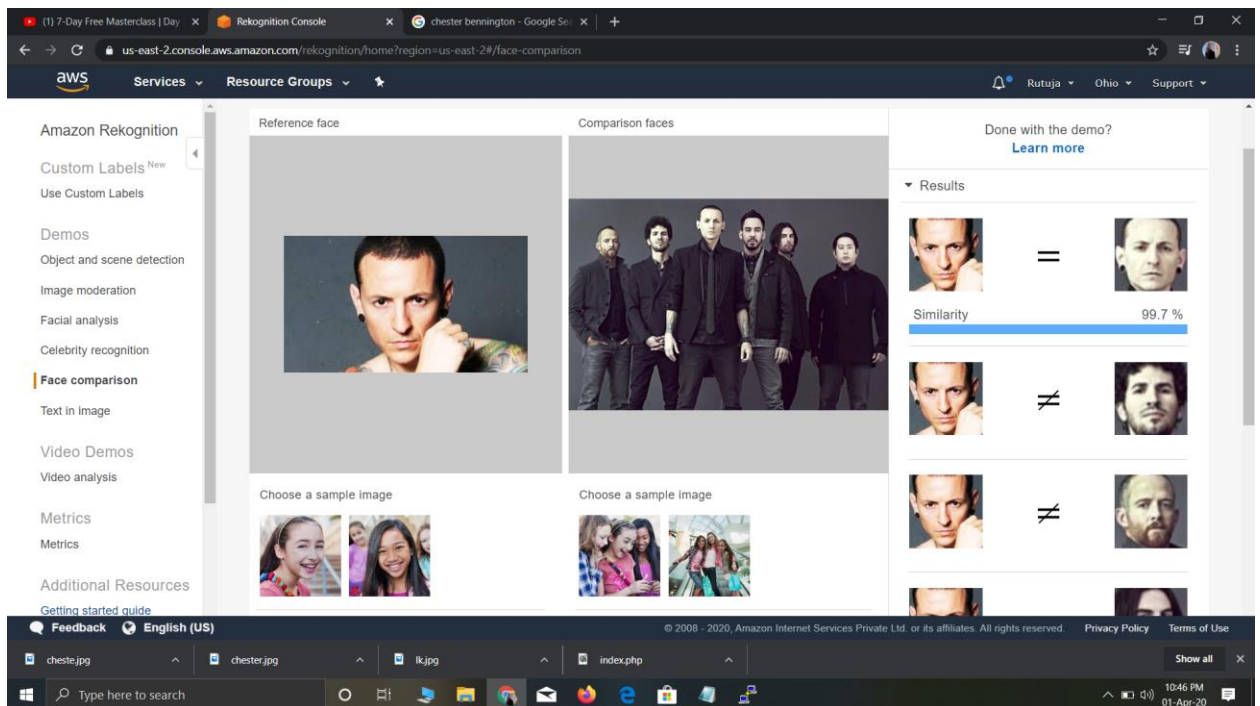


Screenshots needed for Rekognition

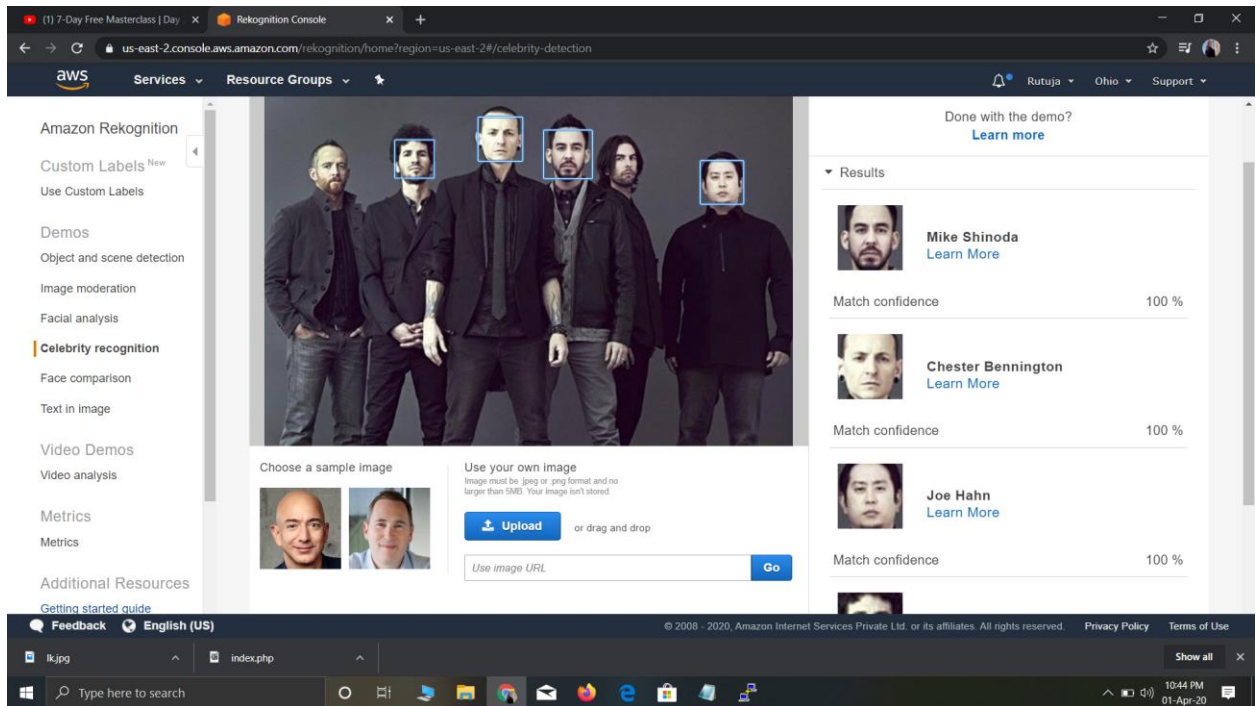
1. Face Detect



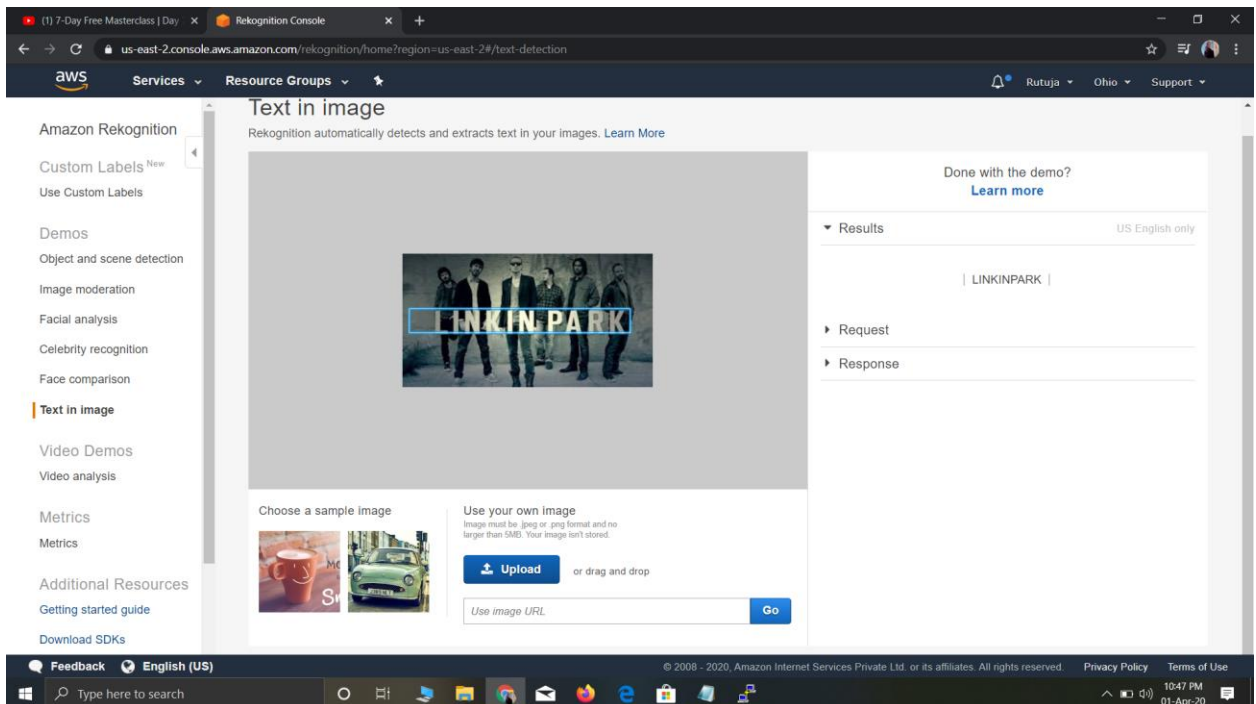
2. Face Compare



3. Celebrity Recognition



4. Text in Image



Screenshots needed for EC2 & S3

1. Installing aws-sdk

```
ec2-user@ip-172-31-35-213:/var/www/html/face
kernel.x86_64 0:4.14.173-137.228.amzn2

Updated:
  gmp2.x86_64 0:2.0.22-5.amzn2.0.4
  langtable.noarch 0:0.0.31-4.amzn2
  langtable-data.noarch 0:0.0.31-4.amzn2
  langtable-python.noarch 0:0.0.31-4.amzn2
  libfastjson.x86_64 0:0.99.4-3.amzn2
  libtirpc.x86_64 0:0.2.4-0.16.amzn2

Complete!
[ec2-user@ip-172-31-35-213 ~]$ sudo yum upgrade
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
No packages marked for update
[ec2-user@ip-172-31-35-213 ~]$ sudo yum install php
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Package php-5.4.16-46.amzn2.0.2.x86_64 already installed and latest version
Nothing to do
[ec2-user@ip-172-31-35-213 ~]$ 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-172-31-35-213 ~]$ cd /var/www/html
[ec2-user@ip-172-31-35-213 html]$ cd face
[ec2-user@ip-172-31-35-213 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 updated
Loading composer repositories with package information
Updating dependencies (including require-dev)
Nothing to install or update
Package guzzle/guzzle is abandoned, you should avoid using it. Use guzzlehttp/guzzle instead.
Generating autoload files
[ec2-user@ip-172-31-35-213 face]$ ^C
[ec2-user@ip-172-31-35-213 face]$ ls
composer.json  linkin-park-billboard-650-compressed.jpg  s.jpg
composer.lock  Linkin-Park.jpg?v=980                      vendor
index.php      sample.jpg
[ec2-user@ip-172-31-35-213 face]$ sudo php index.php
PHP Catchable fatal error:  Argument 1 passed to Aws\Common\Client\AbstractClient::__construct() must be an instance of Aws\Common\Credentials\CredentialsInterface, array given, called in /var/www/html/face/index.php on line 46 and defined in /var/www/html/face/vendor/aws/aws-sdk-php/src/Aws/Common/Client/AbstractClient.php on line 73
[ec2-user@ip-172-31-35-213 face]$
```

2. Installing php

```
ec2-user@ip-172-31-35-213:/var/www/html/face
kernel.x86_64 0:4.14.173-137.228.amzn2

Updated:
  gmp2.x86_64 0:2.0.22-5.amzn2.0.4
  langtable.noarch 0:0.0.31-4.amzn2
  langtable-data.noarch 0:0.0.31-4.amzn2
  langtable-python.noarch 0:0.0.31-4.amzn2
  libfastjson.x86_64 0:0.99.4-3.amzn2
  libtirpc.x86_64 0:0.2.4-0.16.amzn2

Complete!
[ec2-user@ip-172-31-35-213 ~]$ sudo yum upgrade
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
No packages marked for update
[ec2-user@ip-172-31-35-213 ~]$ sudo yum install php
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Package php-5.4.16-46.amzn2.0.2.x86_64 already installed and latest version
Nothing to do
[ec2-user@ip-172-31-35-213 ~]$ 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-172-31-35-213 ~]$ cd /var/www/html
[ec2-user@ip-172-31-35-213 html]$ cd face
[ec2-user@ip-172-31-35-213 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 updated
Loading composer repositories with package information
Updating dependencies (including require-dev)
Nothing to install or update
Package guzzle/guzzle is abandoned, you should avoid using it. Use guzzlehttp/guzzle instead.
Generating autoload files
[ec2-user@ip-172-31-35-213 face]$ ^C
[ec2-user@ip-172-31-35-213 face]$ ls
composer.json  linkin-park-billboard-650-compressed.jpg  s.jpg
composer.lock  Linkin-Park.jpg?v=980                      vendor
index.php      sample.jpg
[ec2-user@ip-172-31-35-213 face]$ sudo php index.php
PHP Catchable fatal error:  Argument 1 passed to Aws\Common\Client\AbstractClient::__construct() must be an instance of Aws\Common\Credentials\CredentialsInterface, array given, called in /var/www/html/face/index.php on line 46 and defined in /var/www/html/face/vendor/aws/aws-sdk-php/src/Aws/Common/Client/AbstractClient.php on line 73
[ec2-user@ip-172-31-35-213 face]$
```

3. index.php file code

```
ec2-user@ip-172-31-40-112/var/www/html/face
$ cat Aws\Rekognition\RekognitionClient.php

$bucket = 'aws-web-07';
$keyname = 's.jpg';

$s3 = S3Client::factory([
    'region' => 'us-east-2',
    'version' => '2006-03-01',
    'signature' => 'v4'
]);

try {
    // Upload image
    $result = $s3->putObject([
        'Bucket' => $bucket,
        'Key' => $keyname,
        'SourceFile' => __DIR__ . "/$keyname",
        'ACL' => 'public-read-write'
    ]);

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

        $rekognition = RekognitionClient::factory([
            'region' => 'us-east-2',
            'version' => 'latest',
        ]);

        $result = $rekognition->detectFaces([
            'Attributes' => ['ALL'],
            'Image' => [
                'S3Object' => [
                    'Bucket' => $bucket,
                    'Name' => $keyname,
                    'Key' => $keyname,
                ],
            ],
        ]);

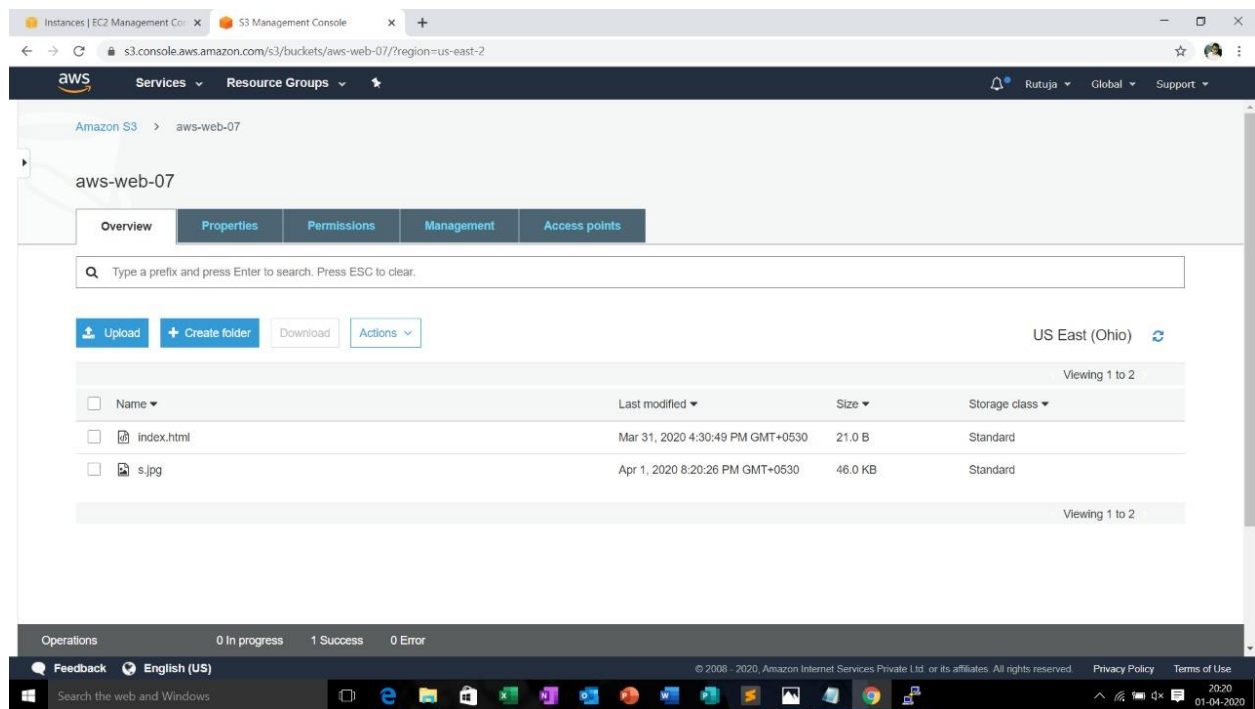
        echo "Totally there are " . count($result['FaceDetails']) . " faces";
    }
} catch (Exception $e) {
    echo $e->getMessage() . PHP_EOL;
}
```

4. Upload success screenshot

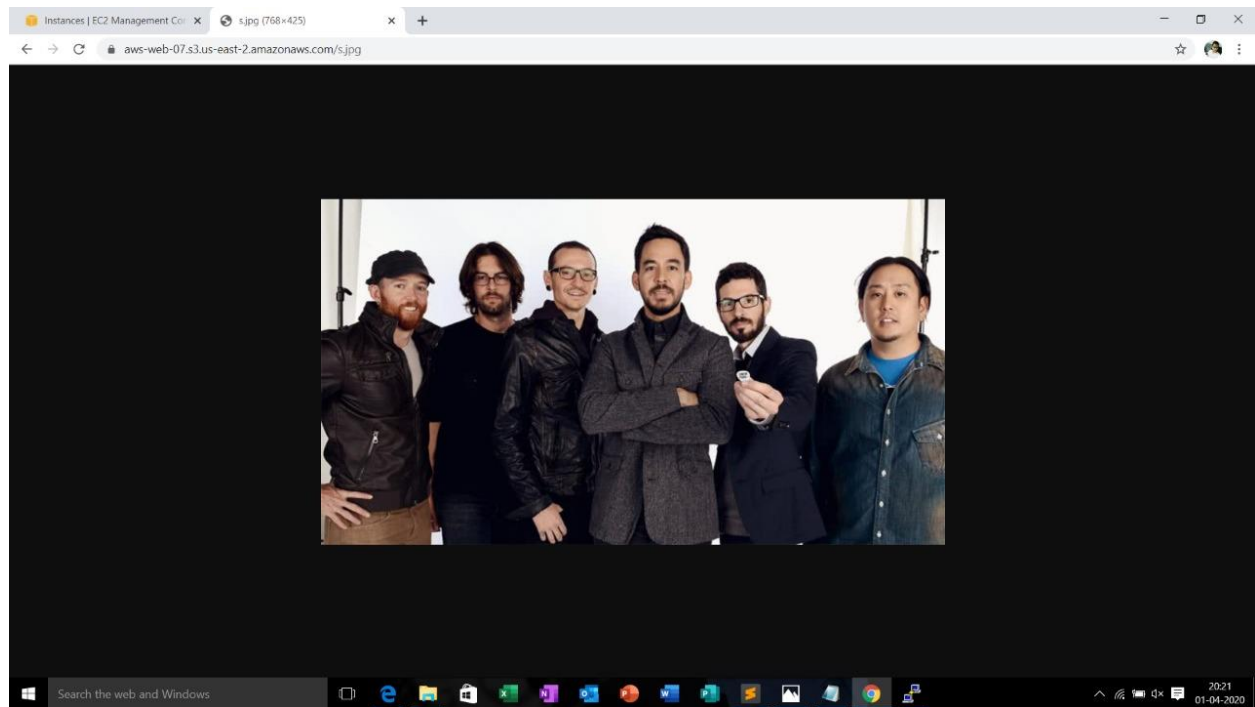
(I)

```
ec2-user@ip-172-31-40-112/var/www/html/face
[ec2-user@ip-172-31-40-112 face]$ sudo vim index.php
[ec2-user@ip-172-31-40-112 face]$ sudo php index.php
Image upload done... Here is the URL: https://aws-web-07.s3.us-east-2.amazonaws.com/s.jpg[ec2-user@ip-172-31-40-112 face]$
```

(II)

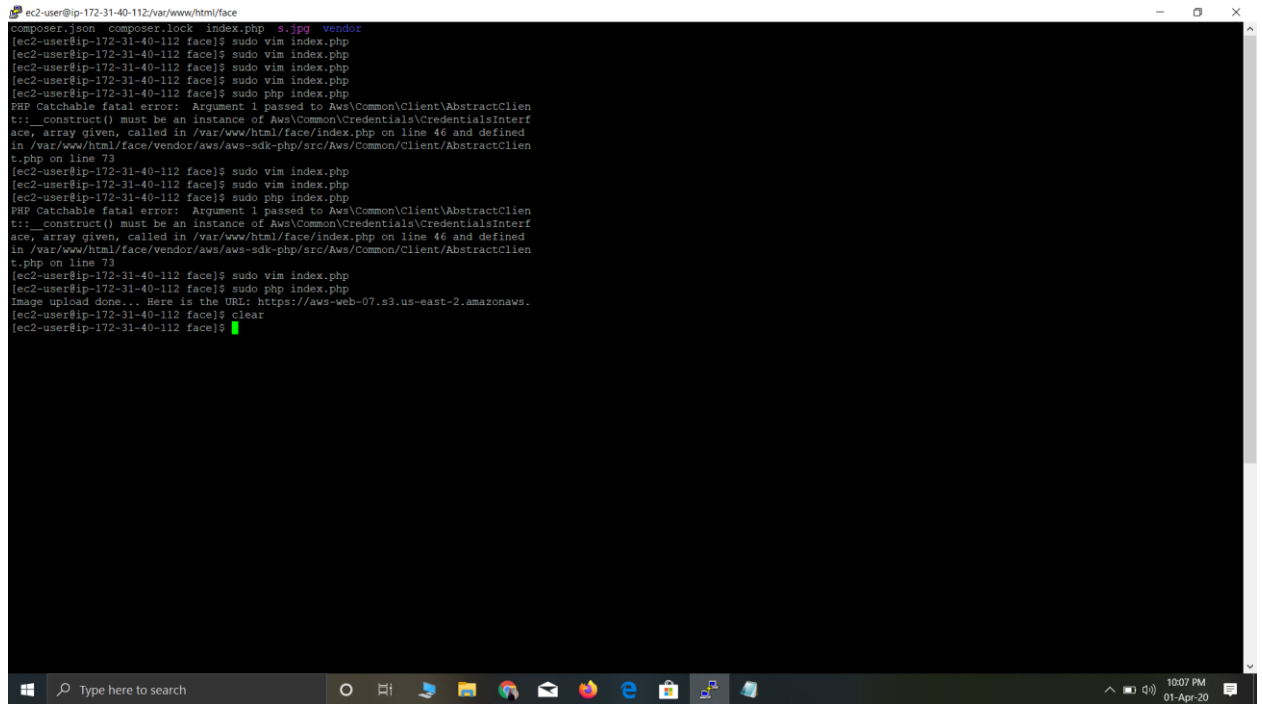


(III)



Screenshots needed for EC2 & Rekognition

1. Face Detect success screenshot



```
ec2-user@ip-172-31-40-112:/var/www/html/face
composer.json composer.lock index.php 9.jpg vendor
[ec2-user@ip-172-31-40-112 face]$ sudo vim index.php
[ec2-user@ip-172-31-40-112 face]$ sudo vim index.php
[ec2-user@ip-172-31-40-112 face]$ sudo vim index.php
[ec2-user@ip-172-31-40-112 face]$ sudo vim index.php
[ec2-user@ip-172-31-40-112 face]$ sudo php index.php
PHP Catchable fatal error: Argument 1 passed to Aws\Common\Client\AbstractClient
::__construct() must be an instance of Aws\Common\Credentials\CredentialsInterf
ace, array given, called in /var/www/html/face/index.php on line 46 and defined
in /var/www/html/face/vendor/aws/aws-sdk-php/src/Aws/Common/Client/AbstractClie
nt.php on line 73
[ec2-user@ip-172-31-40-112 face]$ sudo vim index.php
[ec2-user@ip-172-31-40-112 face]$ sudo vim index.php
[ec2-user@ip-172-31-40-112 face]$ sudo php index.php
PHP Catchable fatal error: Argument 1 passed to Aws\Common\Client\AbstractClient
::__construct() must be an instance of Aws\Common\Credentials\CredentialsInterf
ace, array given, called in /var/www/html/face/index.php on line 46 and defined
in /var/www/html/face/vendor/aws/aws-sdk-php/src/Aws/Common/Client/AbstractClie
nt.php on line 73
[ec2-user@ip-172-31-40-112 face]$ sudo vim index.php
[ec2-user@ip-172-31-40-112 face]$ sudo php index.php
image upload done... Here is the URL: https://aws-web-07.s3.us-east-2.amazonaws.
[ec2-user@ip-172-31-40-112 face]$ clear
[ec2-user@ip-172-31-40-112 face]$
```