19MCA0202
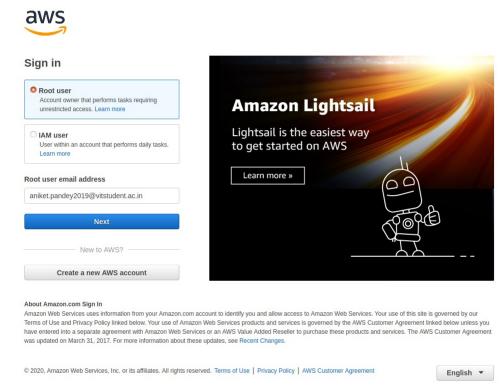**Aniket Pandey**

Screenshots

Screenshots needed for Dashboards

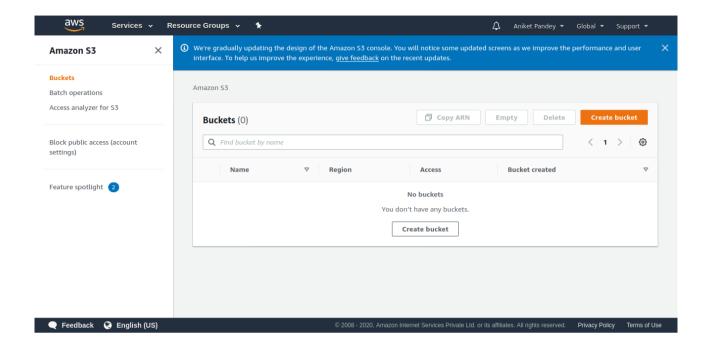## 1. AWS Login screen with username

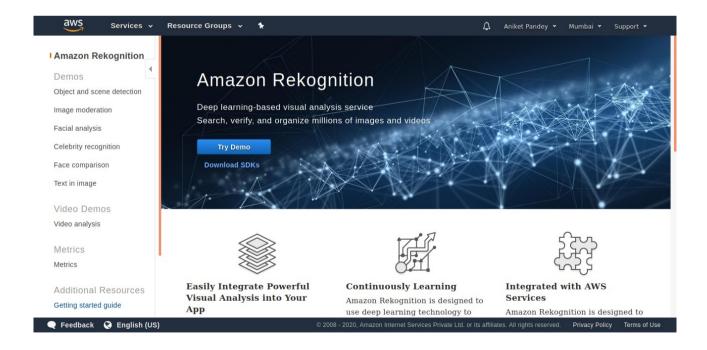aws

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

aniket.pandey2019@vitstudent.ac.in

Next

New to AWS?

Create a new AWS account

Amazon Lightsail

Lightsail is the easiest way to get started on AWS

Learn more »

English ▾

2. EC2 Dashboard

aws   Services ▾   Resource Groups ▾   📌   🔔   Aniket Pandey ▾   Mumbai ▾   Support ▾

🔘 New EC2 Experience
   Tell us what you think

EC2 Dashboard New
Events New
Tags
Reports
Limits

▼ **INSTANCES**
   Instances
   Instance Types
   Launch Templates New
   Spot Requests
   Savings Plans
   Reserved Instances
   Dedicated Hosts New
   Capacity Reservations

▼ **IMAGES**
   AMIs

ⓘ **Welcome to the new EC2 console!**
We're redesigning the EC2 console to make it easier to use and improve performance. We'll release new screens periodically. We encourage you to try them and let us know where we can make improvements. To switch between the old console and the new console, use the New EC2 Experience toggle.   ✕

EC2

Resources   ↻ ⚙

You are using the following Amazon EC2 resources in the Asia Pacific (Mumbai) Region:

Running instances	0	**Elastic IPs**	0
Dedicated Hosts	0	**Snapshots**	0
Volumes	0	**Load balancers**	0
Key pairs	1	**Security groups**	2
Placement groups	0		

ⓘ Easily size, configure, and deploy Microsoft SQL Server Always On availability groups   ✕

Account attributes
↻

Supported platforms 🔗
• VPC

Default VPC 🔗
vpc-6d343805

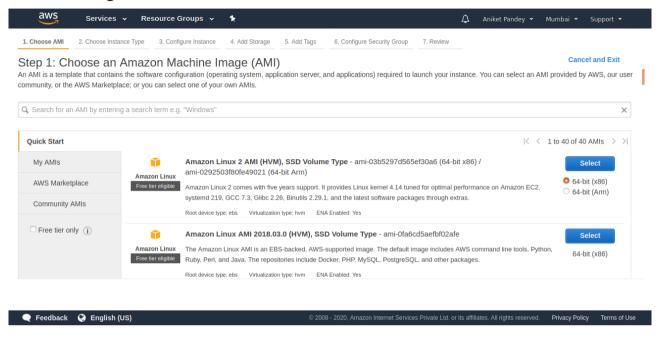Console experiments

Settings

Explore AWS   ✕
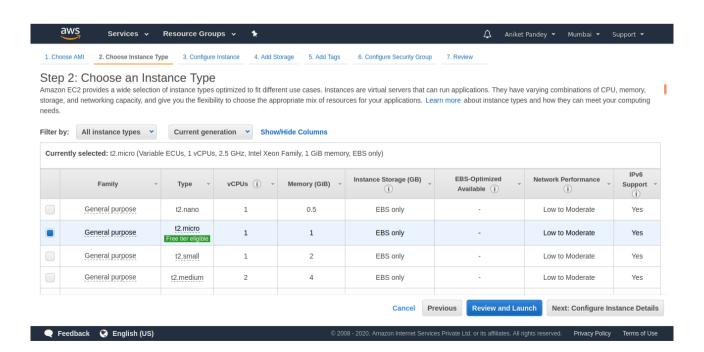
3. S3 Dashboard

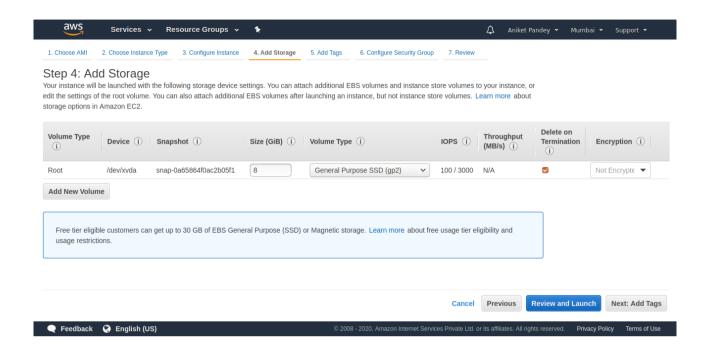
4. Rekognition Dashboard

Screenshots needed for EC2

1. Choosing an AMI
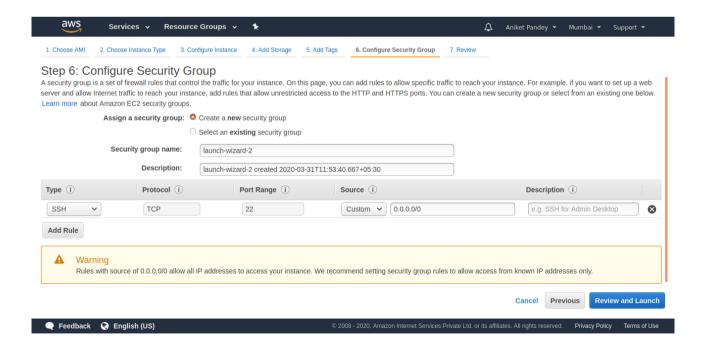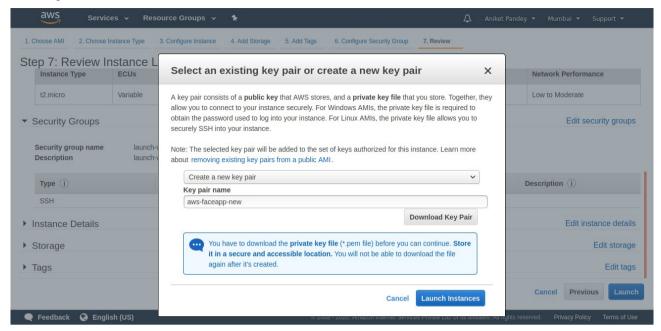

2. Choosing an Instance Type
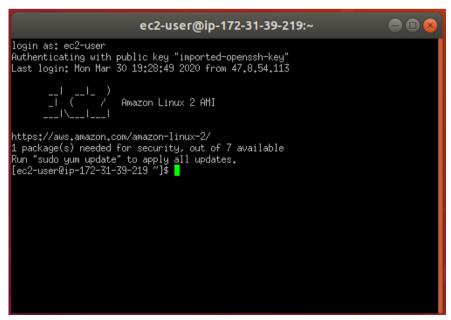
3. Adding Storage

4. Configuring Security Group

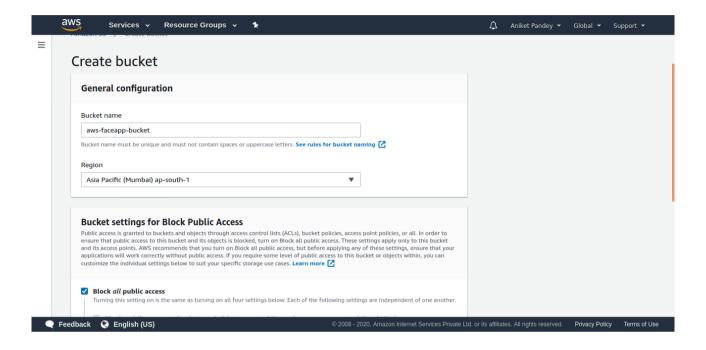## 5. Key Pair Download

6. PuTTYgen conversion from pem to ppk
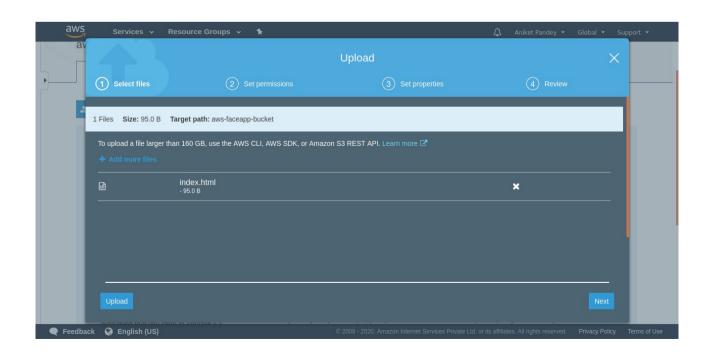

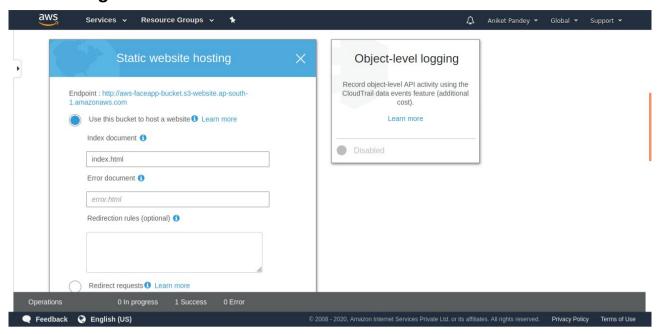## 7. Logged in EC2 black screen

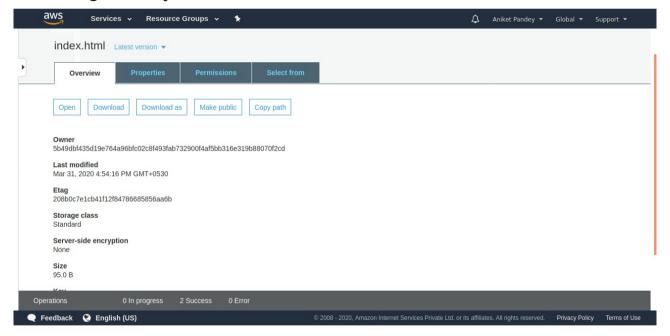# Screenshots needed for S3

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
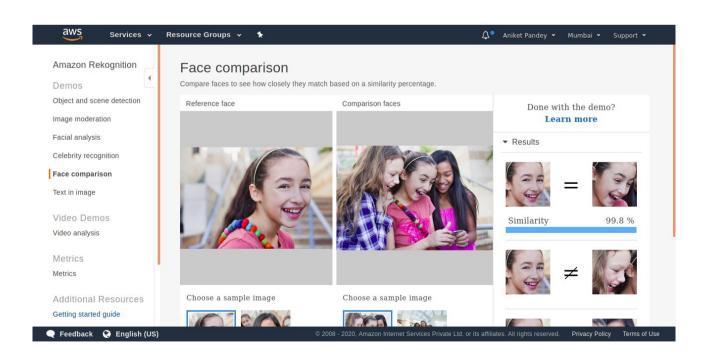

This is a paragraph.

Screenshots needed for Rekognition
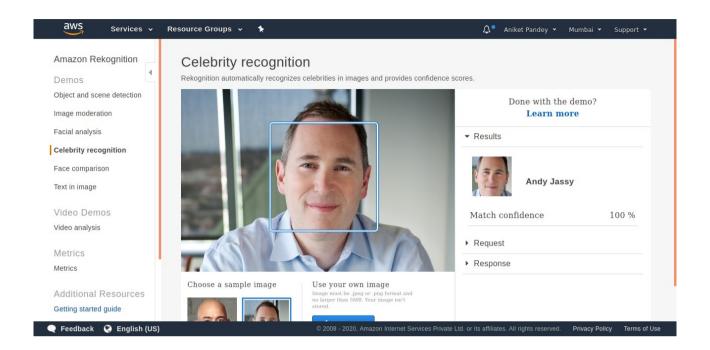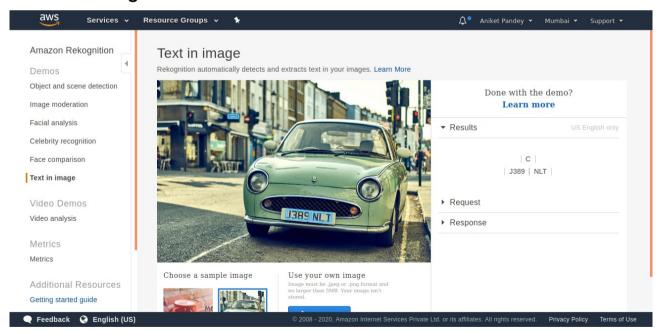
1. Face Detect

2. Face Compare

3. Celebrity Recognition

4. Text in Image

Screenshots needed for EC2 & S3

1. Installing aws-sdk

```
[ec2-user@ip-172-31-44-55 face-app]$ 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
```

2. Installing php

```
[ec2-user@ip-172-31-44-55 ~]$ 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: ph
p-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
--> 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
```

3. index.php file code

```php
<?php
require_once(__DIR__ . '/vendor/autoload.php');

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


$bucket = 'aws-faceapp-ethnus';
$keyname = 'sample.jpg';

$s3 = S3Client::factory([
        'region'          => 'ap-south-1',
        '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'          => 'ap-south-1',
                        '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-172-31-46-79 face]$ sudo php index.php
PHP Warning:  strtotime(): It is not safe to rely on the system's timezone settings. You are *required*
thods and you are still getting this warning, you most likely misspelled the timezone identifier. We se
vendor/aws/aws-sdk-php/src/Aws/Common/InstanceMetadata/InstanceMetadataClient.php on line 99
Image upload done... Here is the URL: https://aws-faceapp-ethnus.s3.ap-south-1.amazonaws.com/sample.jpg
2
[ec2-user@ip-172-31-46-79 face]$ cd vendor
```

Screenshots needed for EC2 & Rekognition

1. Face Detect success screenshot

```
[ec2-user@ip-172-31-46-79 face]$ sudo php index.php
Image upload done... Here is the URL: https://aws-faceapp-ethnus.s3.ap-south-1.amazonaws.com/sample.jpgTotally there are 9 faces[ec2-user@ip-172-31-46-79 face]$
```