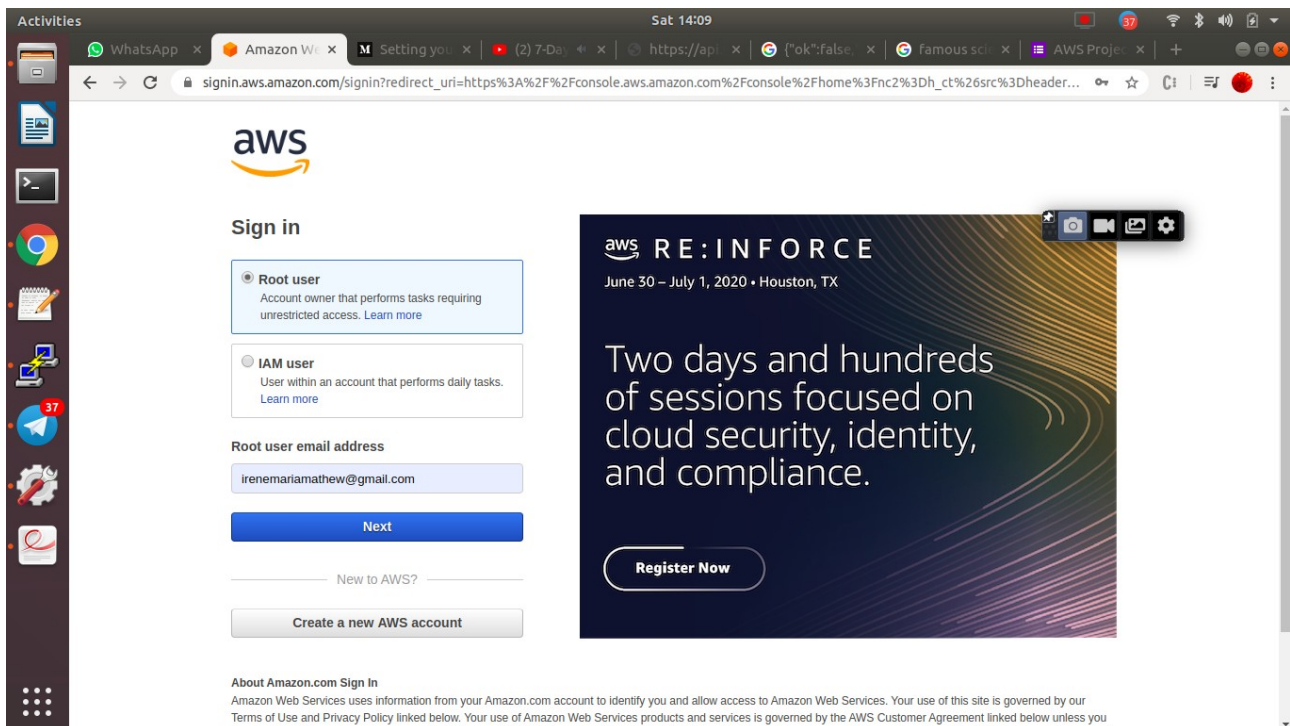


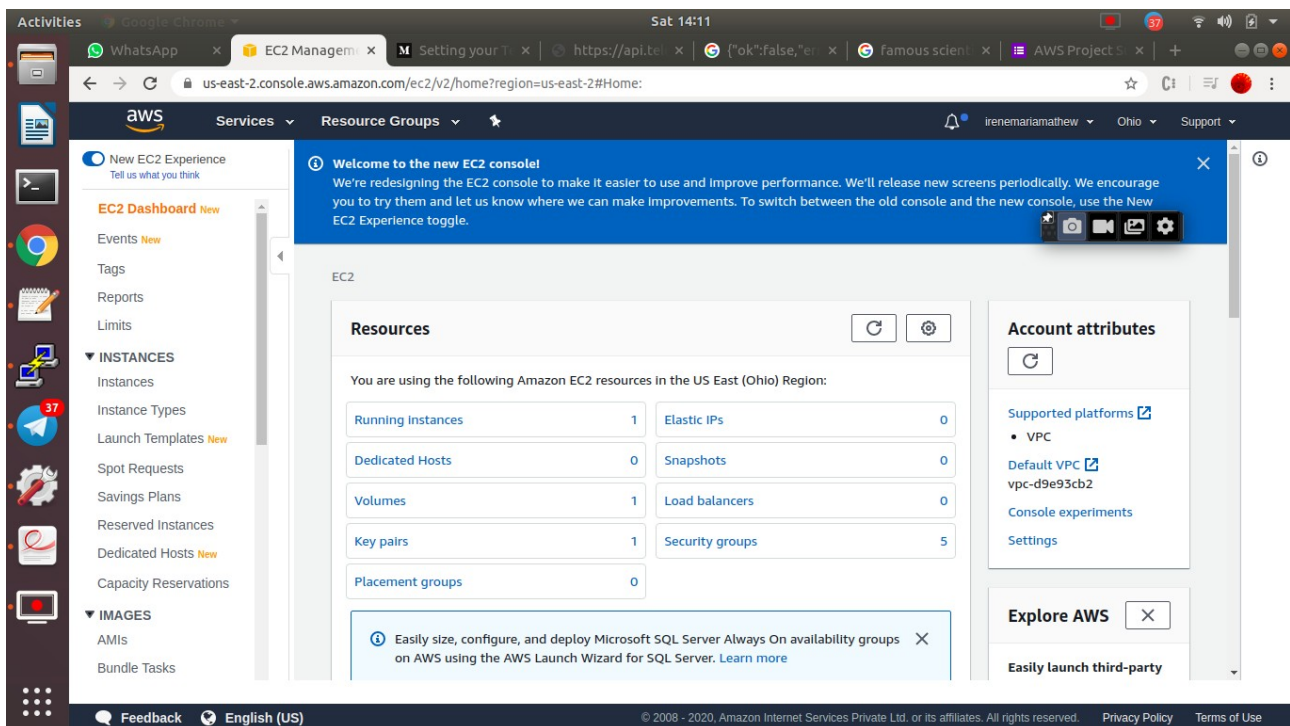
# AWS FACE DETECTION

Screenshots needed for Dashboards

## 1. AWS Login screen with username



## 2. EC2 Dashboard



### 3. S3 Dashboard

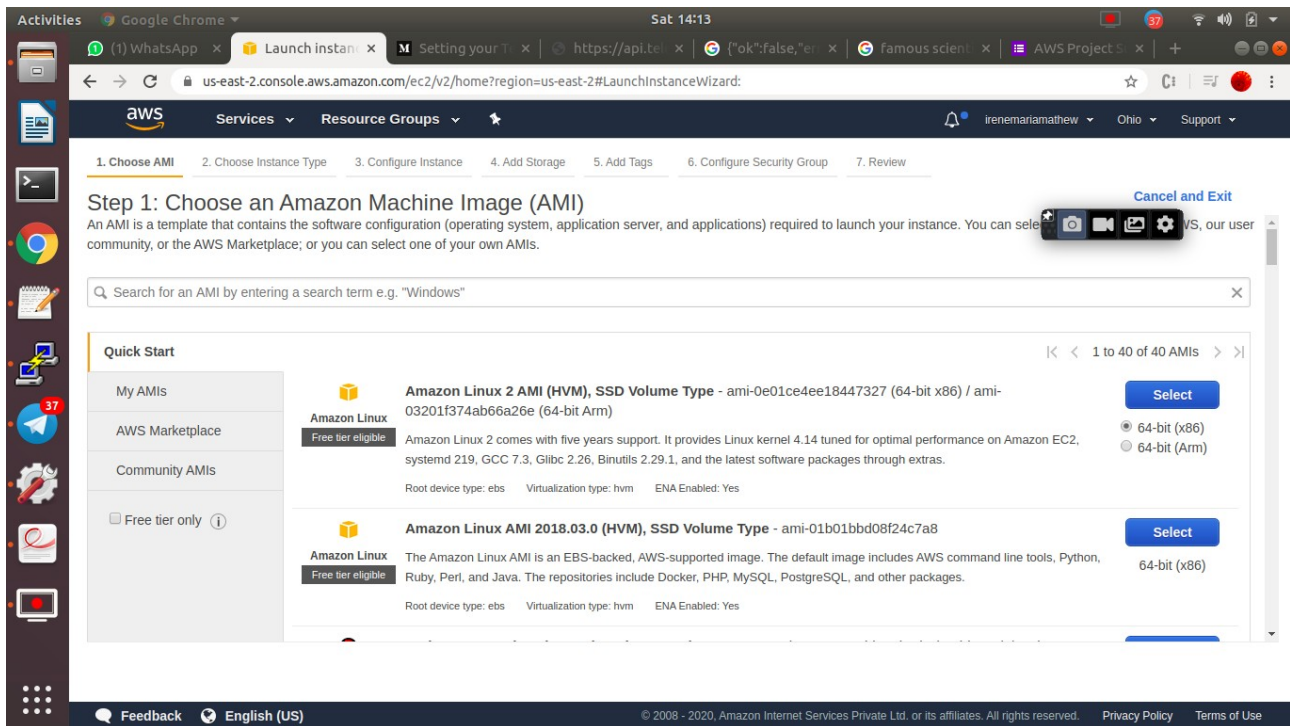
The screenshot shows the Amazon S3 console in a Google Chrome browser. The URL is `s3.console.aws.amazon.com/s3/home?region=us-east-2#`. The left sidebar contains the 'Amazon S3' menu with options like 'Buckets', 'Batch operations', 'Access analyzer for S3', 'Block public access (account settings)', and 'Feature spotlight'. The main content area displays 'Buckets (1)' with a table listing one bucket: 'aws-webinar-ethnus-1' in the 'US East (Ohio) us-east-2' region, created on '2020-03-30T14:49:34.000Z'. Above the table are buttons for 'Copy ARN', 'Empty', 'Delete', and 'Create bucket'. A search bar 'Find bucket by name' is also present. A blue banner at the top right mentions updates to the S3 console design.

### 4. Rekognition Dashboard

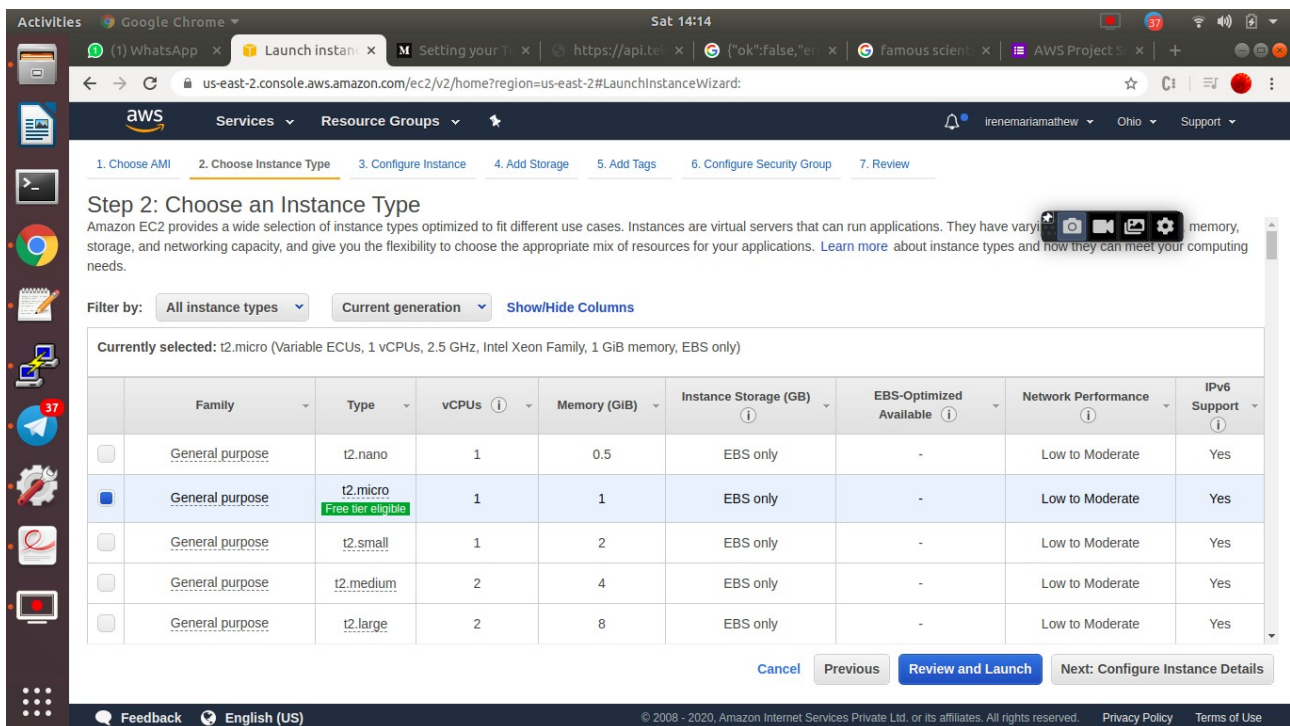
The screenshot shows the Amazon Rekognition console in a Google Chrome browser. The URL is `us-east-2.console.aws.amazon.com/rekognition/home?region=us-east-2#`. The left sidebar contains the 'Amazon Rekognition' menu with options like 'Custom Labels', 'Demos', 'Video Demos', and 'Metrics'. The main content area features a large header with the text 'Amazon Rekognition' and 'Deep learning-based visual analysis service'. Below this is a 'Try Demo' button and a 'Download SDKs' link. The bottom section highlights three key features: 'Easily Integrate Powerful Visual Analysis into Your App', 'Continuously Learning', and 'Integrated with AWS Services'.

## Screenshots needed for EC2

### 1. Choosing an AMI



### 2. Choosing an Instance Type





### 3. Adding Storage

The screenshot shows the AWS Management Console's Launch Instance Wizard at Step 4: Add Storage. The breadcrumb trail at the top indicates the steps: 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage (current), 5. Add Tags, 6. Configure Security Group, and 7. Review. The main heading is "Step 4: Add Storage". Below it, a paragraph explains that the instance will be launched with the following storage device settings and that additional EBS volumes can be attached after launch. A table lists the storage 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 Encrypt

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 buttons for "Cancel", "Previous", "Review and Launch", and "Next: Add Tags".

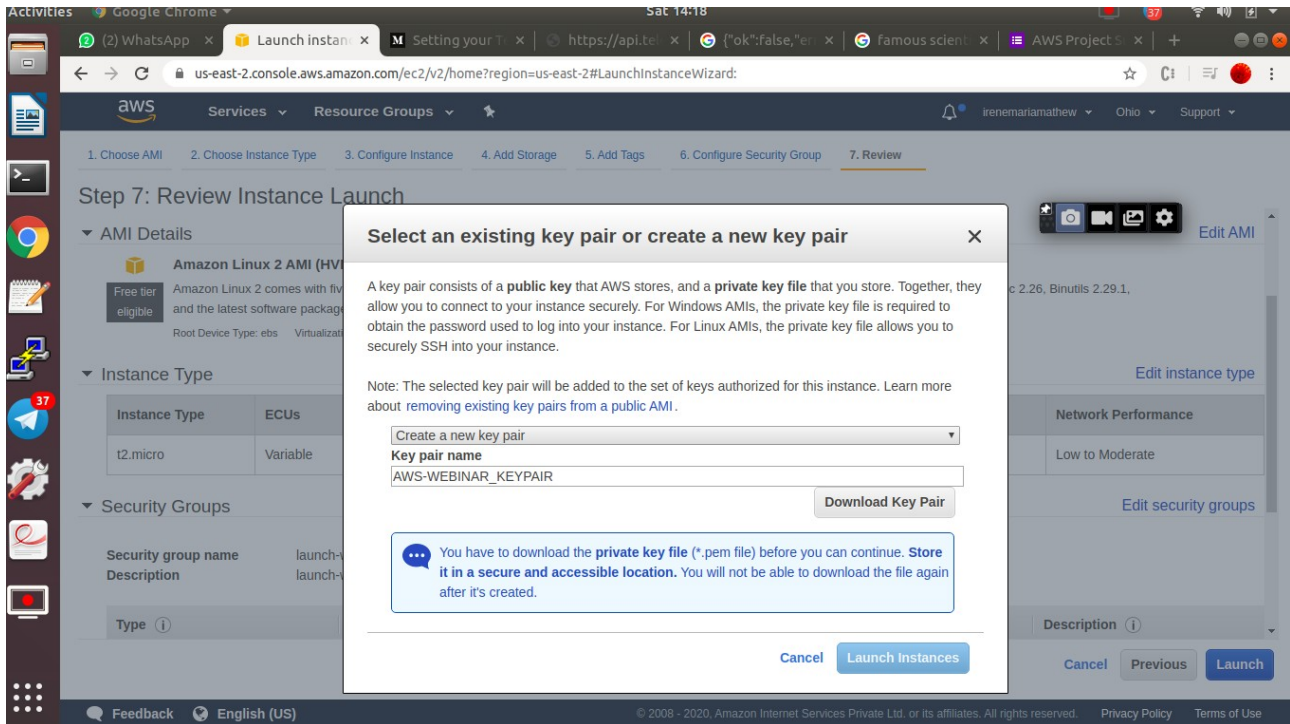
### 4. Configuring Security Group

The screenshot shows the AWS Management Console's Launch Instance Wizard at Step 6: Configure Security Group. The breadcrumb trail at the top indicates the steps: 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage, 5. Add Tags, 6. Configure Security Group (current), and 7. Review. The main heading is "Step 6: Configure Security Group". Below it, a paragraph explains that a security group is a set of firewall rules that control traffic to the instance and that rules can be added to allow specific traffic. A section titled "Assign a security group:" has two radio buttons: "Create a new security group" (selected) and "Select an existing security group". Below this, the "Security group name:" field contains "launch-wizard-5" and the "Description:" field contains "launch-wizard-5 created 2020-04-04T14:16:24.040+05:30".

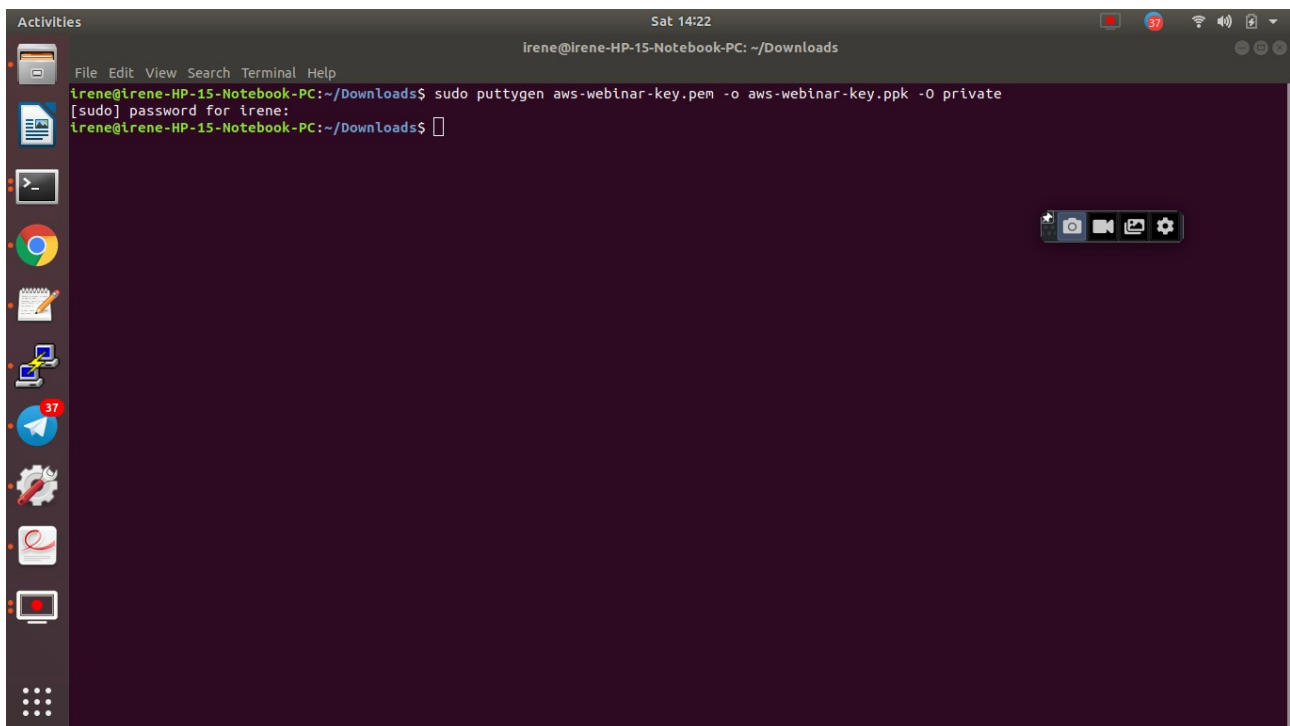
Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
HTTP	TCP	80	Anywhere 0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop

Below the table is an "Add Rule" button. A yellow warning box with a triangle icon 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 buttons for "Cancel", "Previous", "Review and Launch", and "Next: Add Tags".

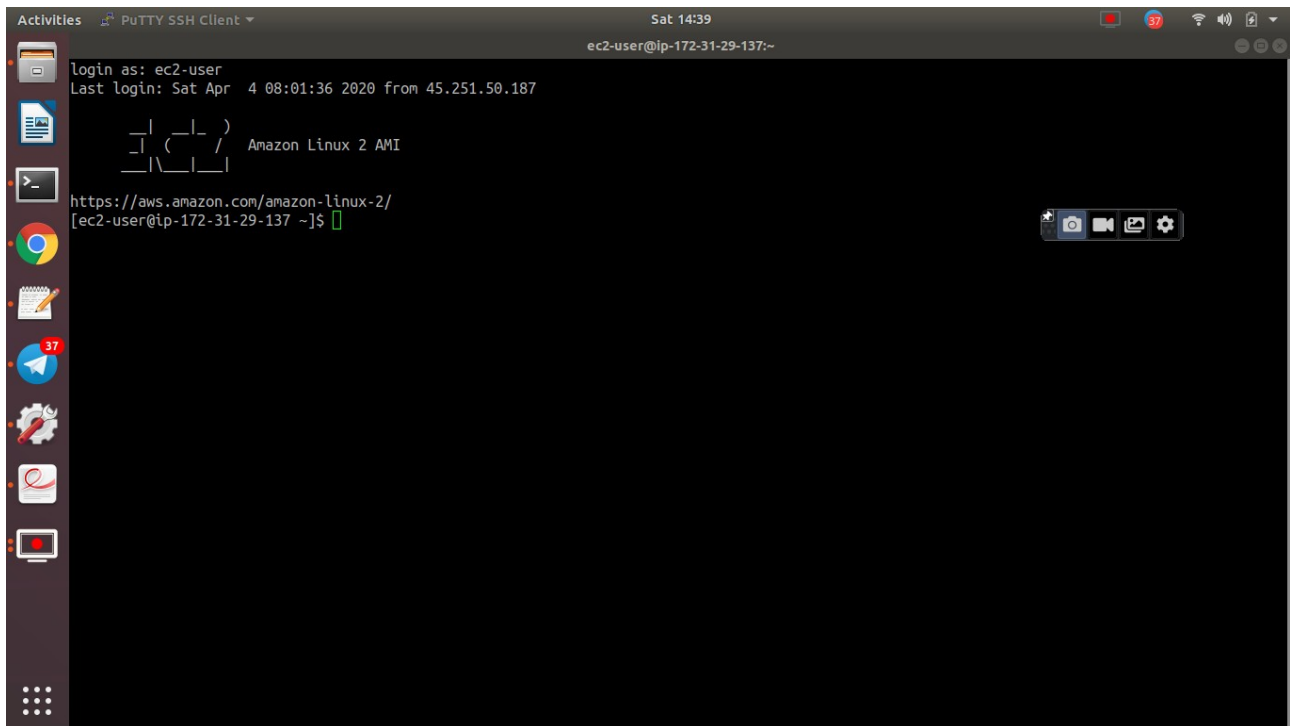
## 5. Key Pair Download



## 6. PuTTYgen conversion from pem to ppk

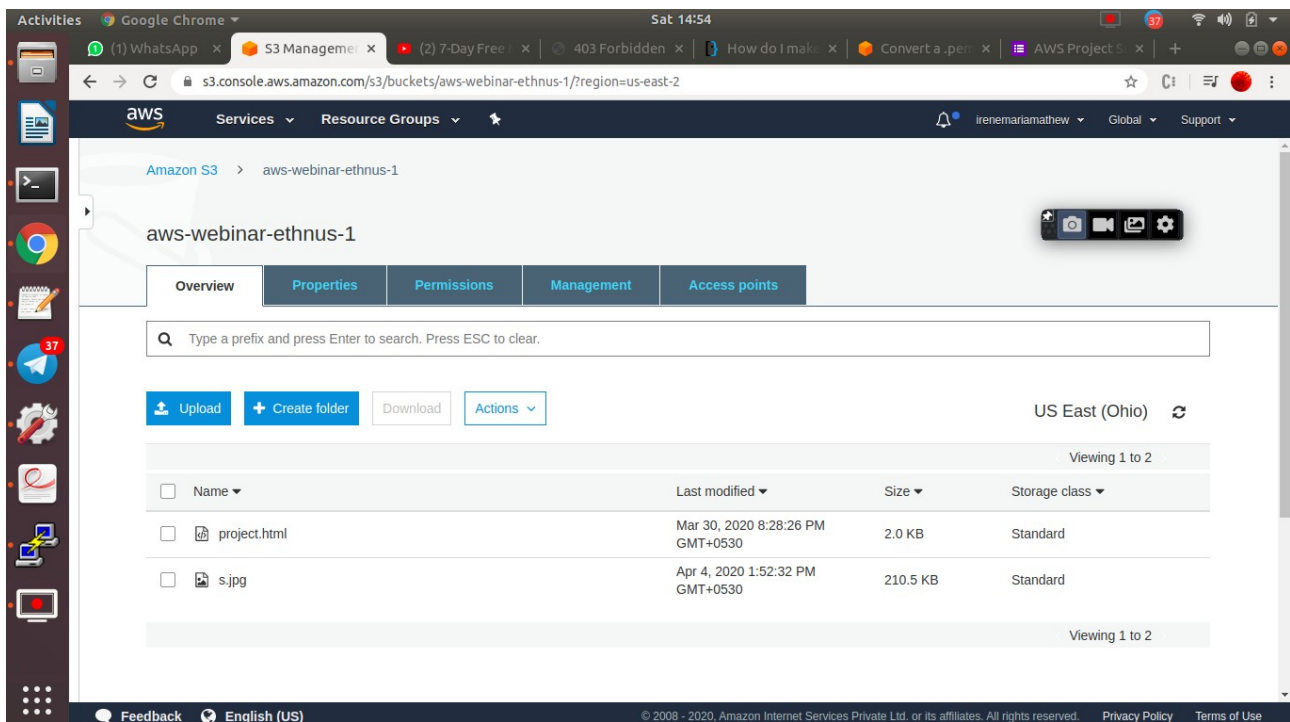


## 7. Logged in EC2 black screen

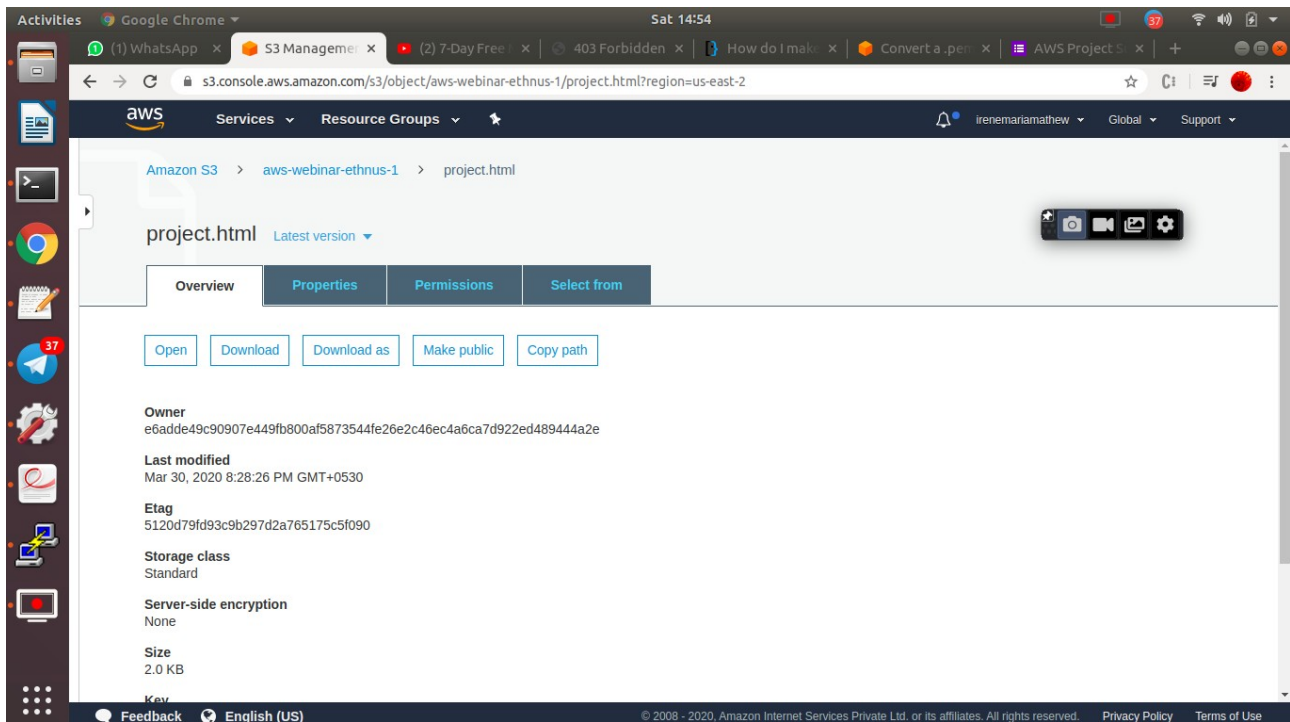


## Screenshots needed for S3

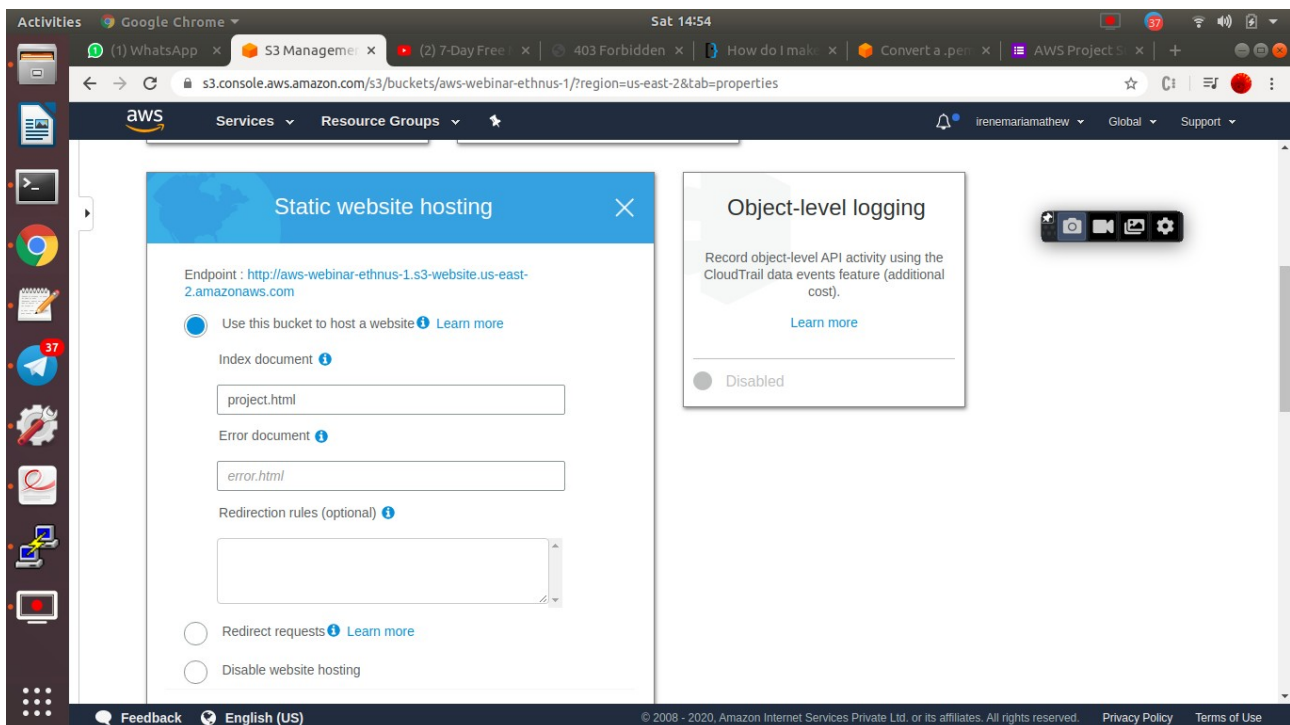
### 1. Creating a bucket



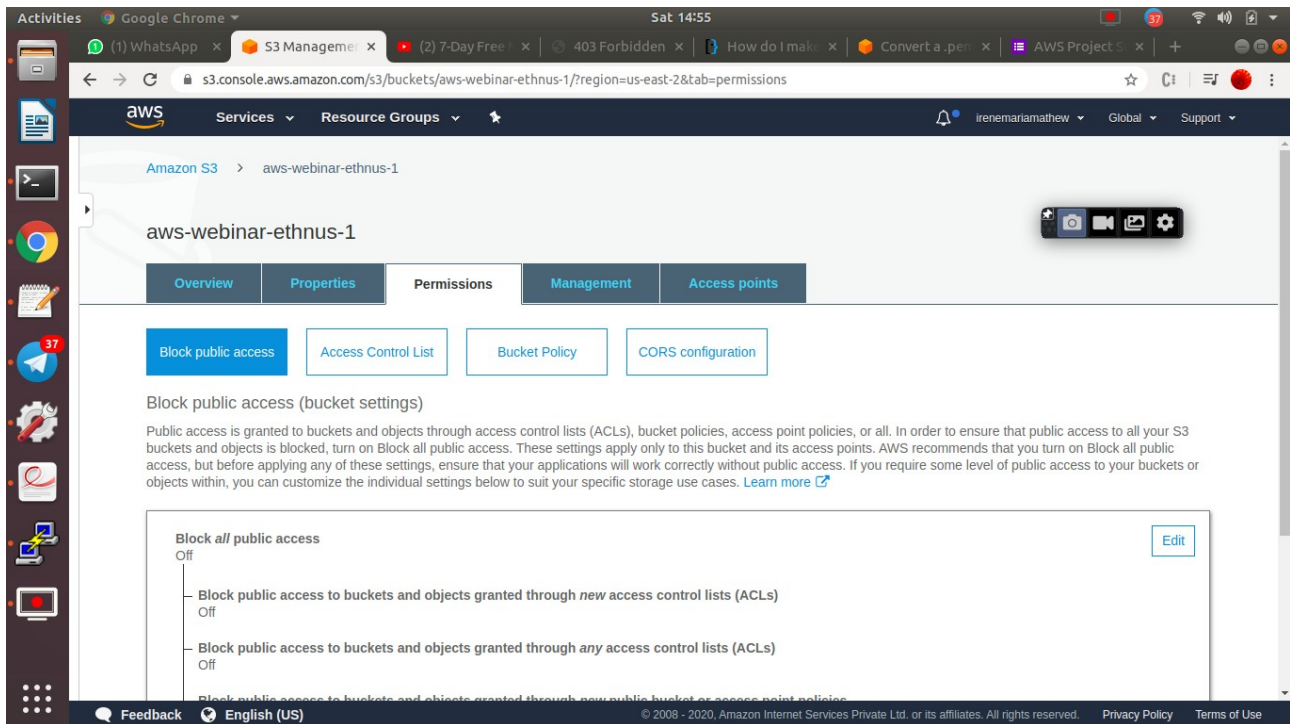
## 2. Uploading an Object



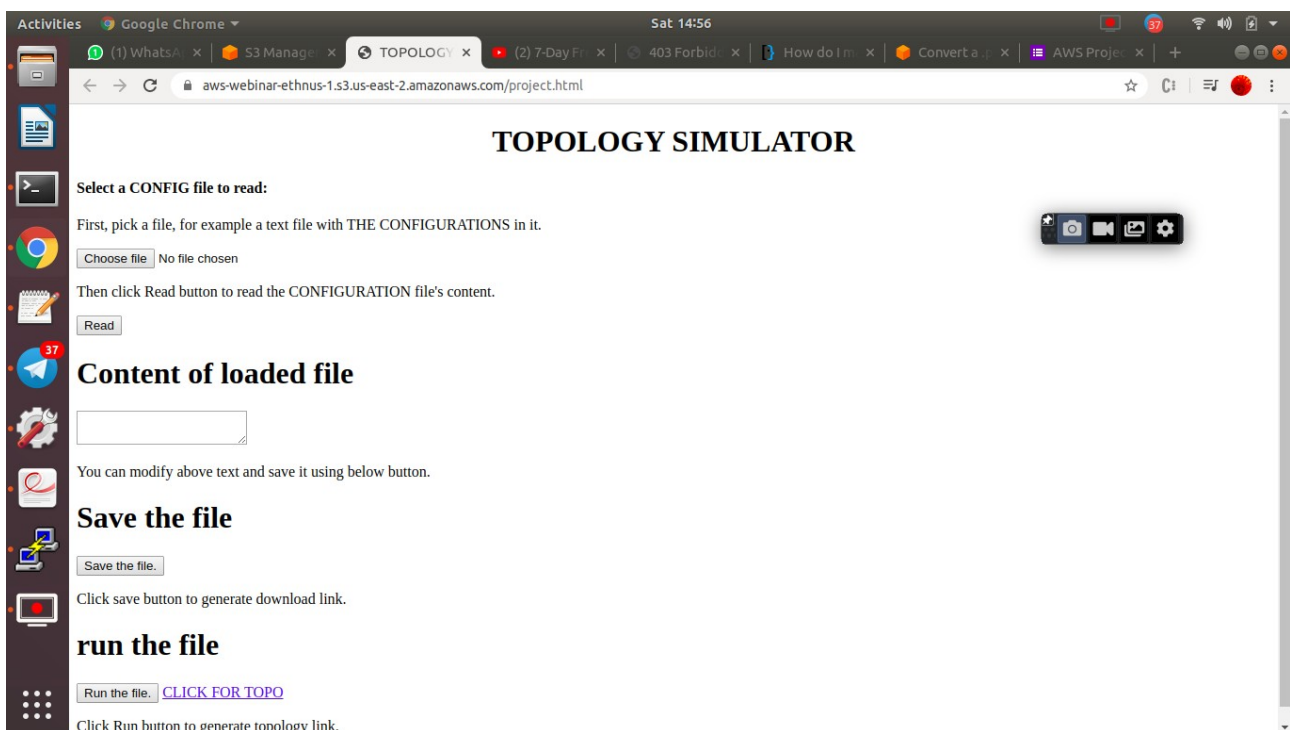
## 3. Enabling Static Website



## 4. Making the Object Public



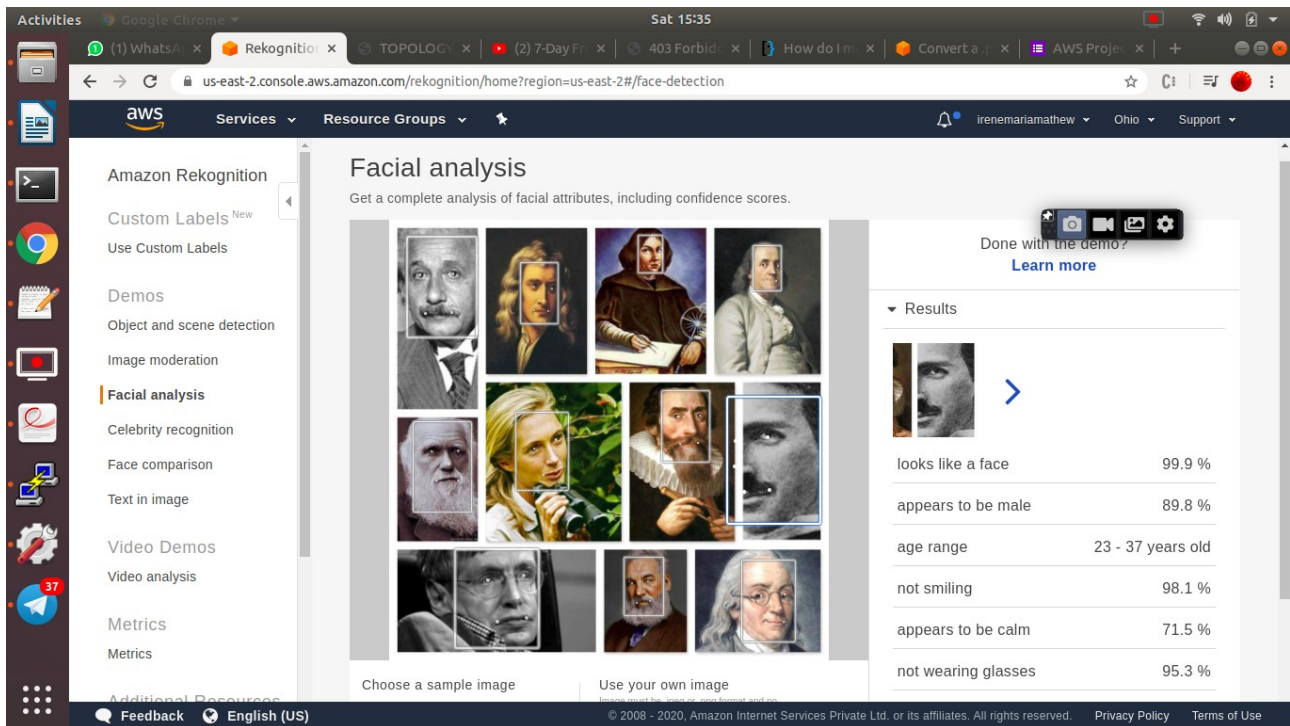
## 5. Checking the S3 link on the browser



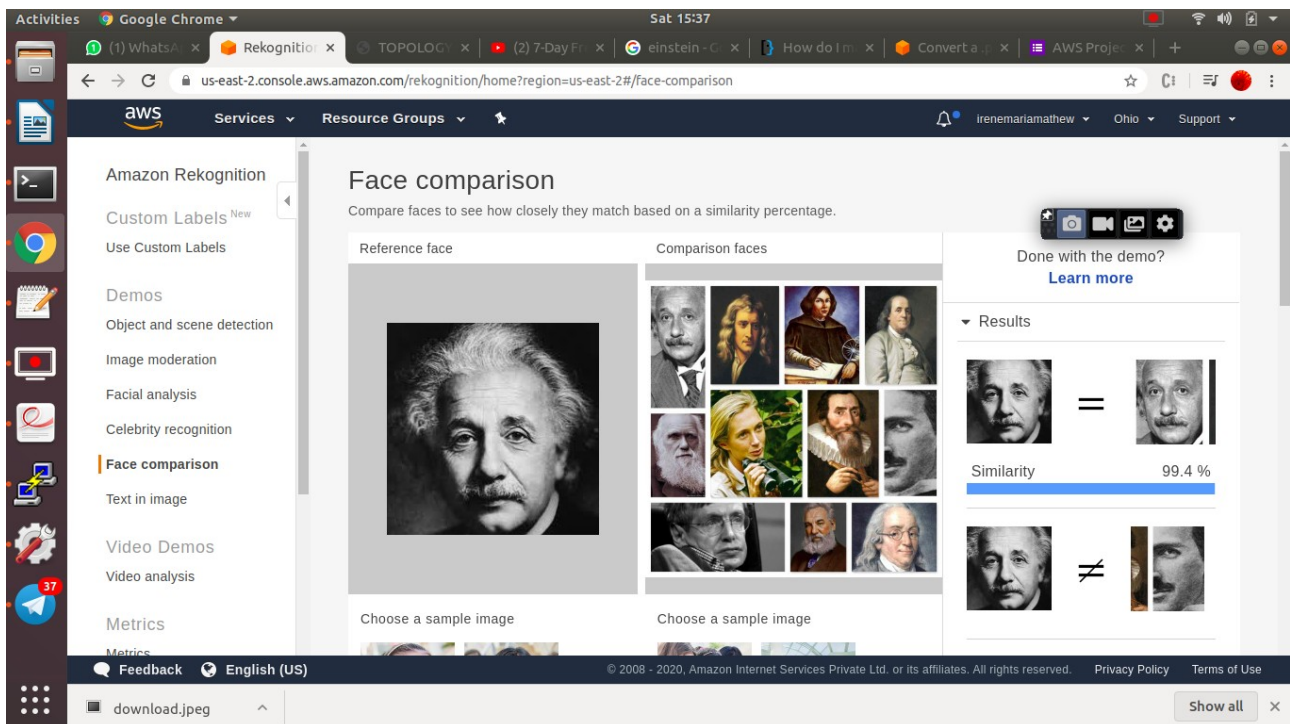


# Screenshots needed for Rekognition

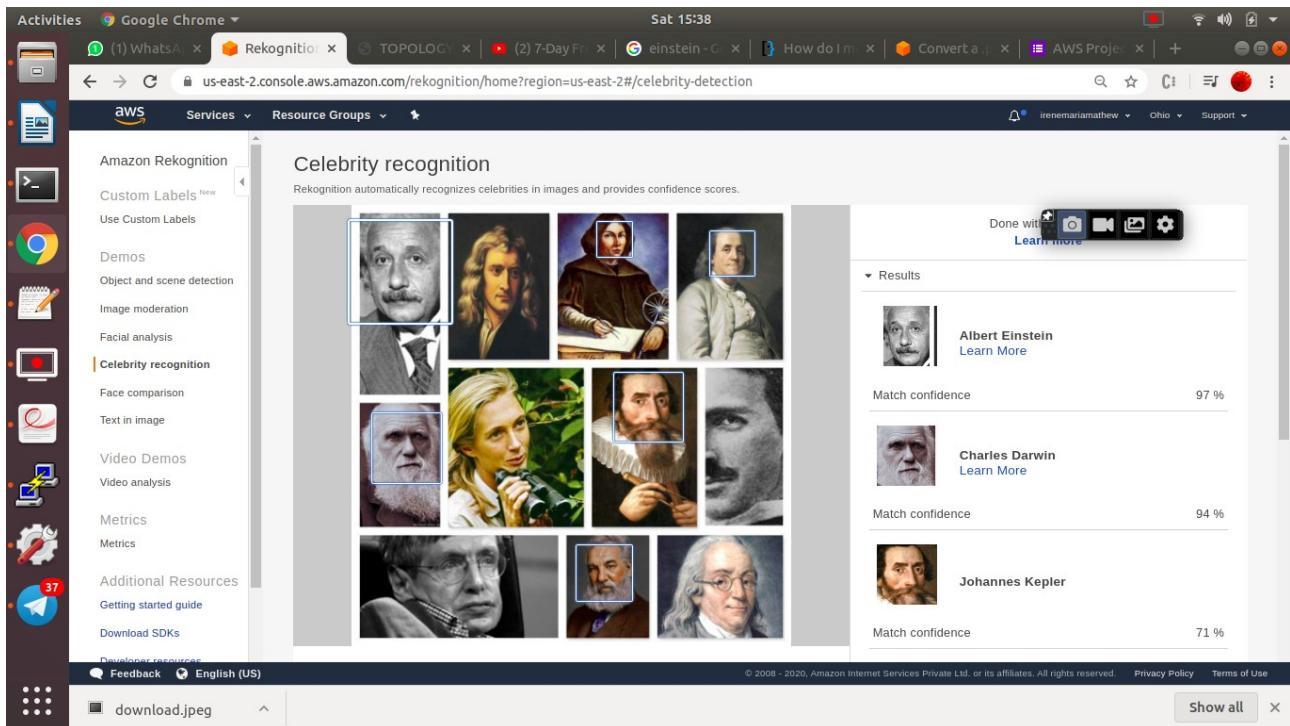
## 1. Face Detect



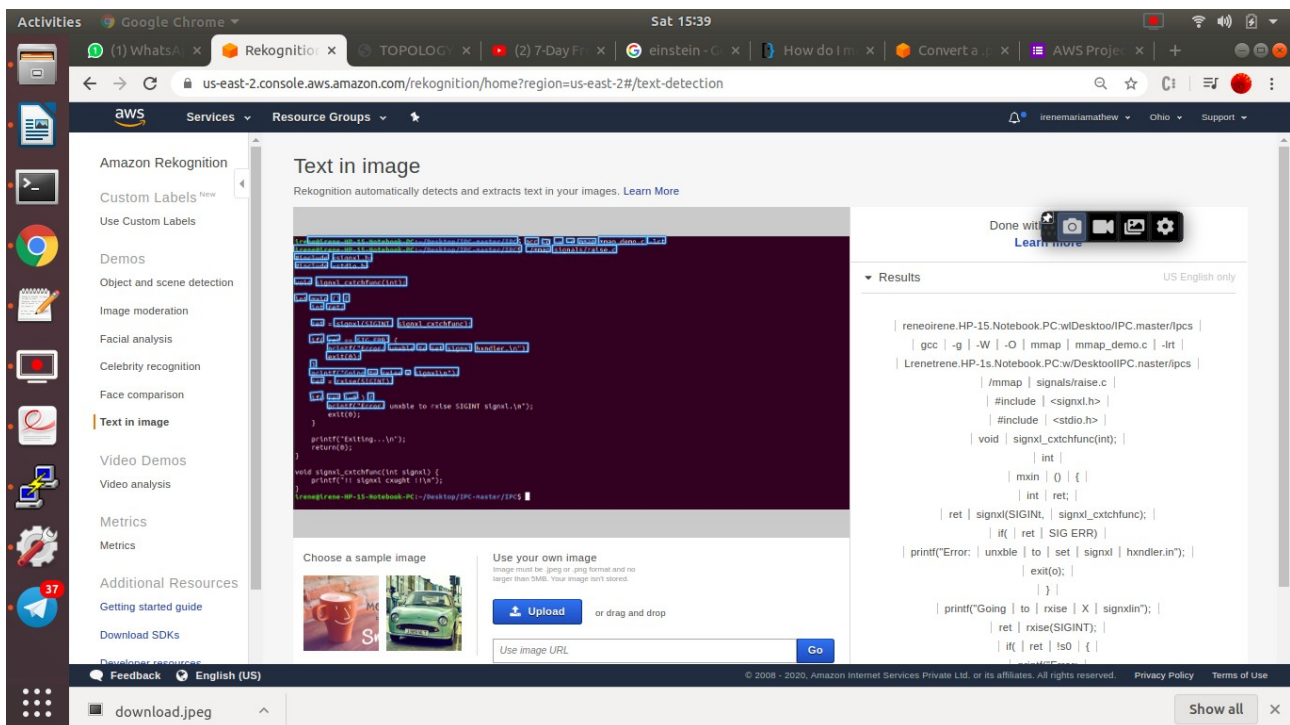
## 2. Face Compare



### 3. Celebrity Recognition

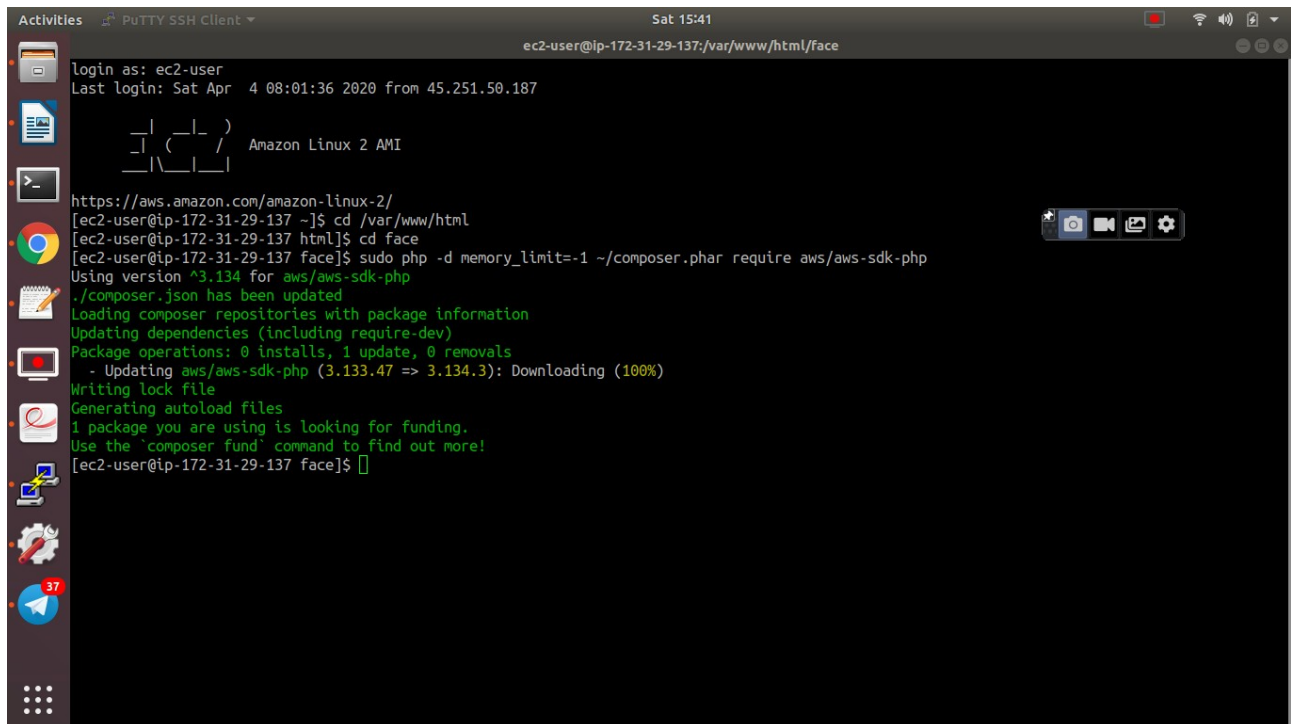


### 4. Text in Image



## Screenshots needed for EC2 & S3

### 1. Installing aws-sdk

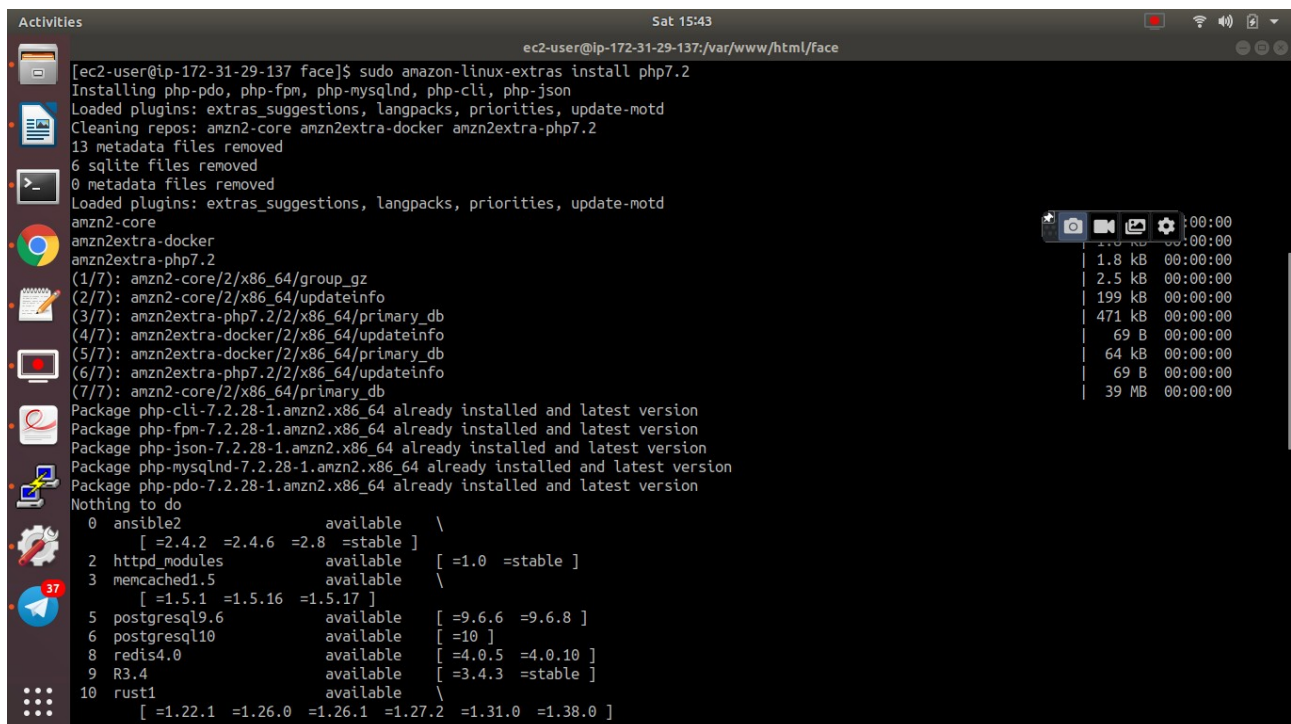


```
login as: ec2-user
Last login: Sat Apr  4 08:01:36 2020 from 45.251.50.187

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

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-29-137 ~]$ cd /var/www/html
[ec2-user@ip-172-31-29-137 html]$ cd face
[ec2-user@ip-172-31-29-137 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 => 3.134.3): Downloading (100%)
Writing 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-172-31-29-137 face]$
```

### 2. Installing php



```
[ec2-user@ip-172-31-29-137 face]$ sudo amazon-linux-extras install php7.2
Installing php-pdo, php-fpm, php-mysqlnd, php-cli, php-json
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Cleaning repos: amzn2-core amzn2extra-docker amzn2extra-php7.2
13 metadata files removed
6 sqlite files removed
0 metadata files removed
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
amzn2extra-docker
amzn2extra-php7.2
(1/7): amzn2-core/2/x86_64/group_gz
(2/7): amzn2-core/2/x86_64/updateinfo
(3/7): amzn2extra-php7.2/2/x86_64/primary_db
(4/7): amzn2extra-docker/2/x86_64/updateinfo
(5/7): amzn2extra-docker/2/x86_64/primary_db
(6/7): amzn2extra-php7.2/2/x86_64/updateinfo
(7/7): amzn2-core/2/x86_64/primary_db
Package php-cli-7.2.28-1.amzn2.x86_64 already installed and latest version
Package php-fpm-7.2.28-1.amzn2.x86_64 already installed and latest version
Package php-json-7.2.28-1.amzn2.x86_64 already installed and latest version
Package php-mysqlnd-7.2.28-1.amzn2.x86_64 already installed and latest version
Package php-pdo-7.2.28-1.amzn2.x86_64 already installed and latest version
Nothing to do
0  ansible2                available      \
   [ =2.4.2 =2.4.6 =2.8  =stable ]
2  httpd_modules            available      [ =1.0  =stable ]
3  memcached1.5             available      \
   [ =1.5.1 =1.5.16 =1.5.17 ]
5  postgresql9.6            available      [ =9.6.6 =9.6.8 ]
6  postgresql10             available      [ =10 ]
8  redis4.0                 available      [ =4.0.5 =4.0.10 ]
9  R3.4                     available      [ =3.4.3 =stable ]
10 rust1                    available      \
   [ =1.22.1 =1.26.0 =1.26.1 =1.27.2 =1.31.0 =1.38.0 ]
```



### 3. index.php file code

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

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

$bucket = 'aws-webinar-ethnus-1';
$keyname = 's.jpg';

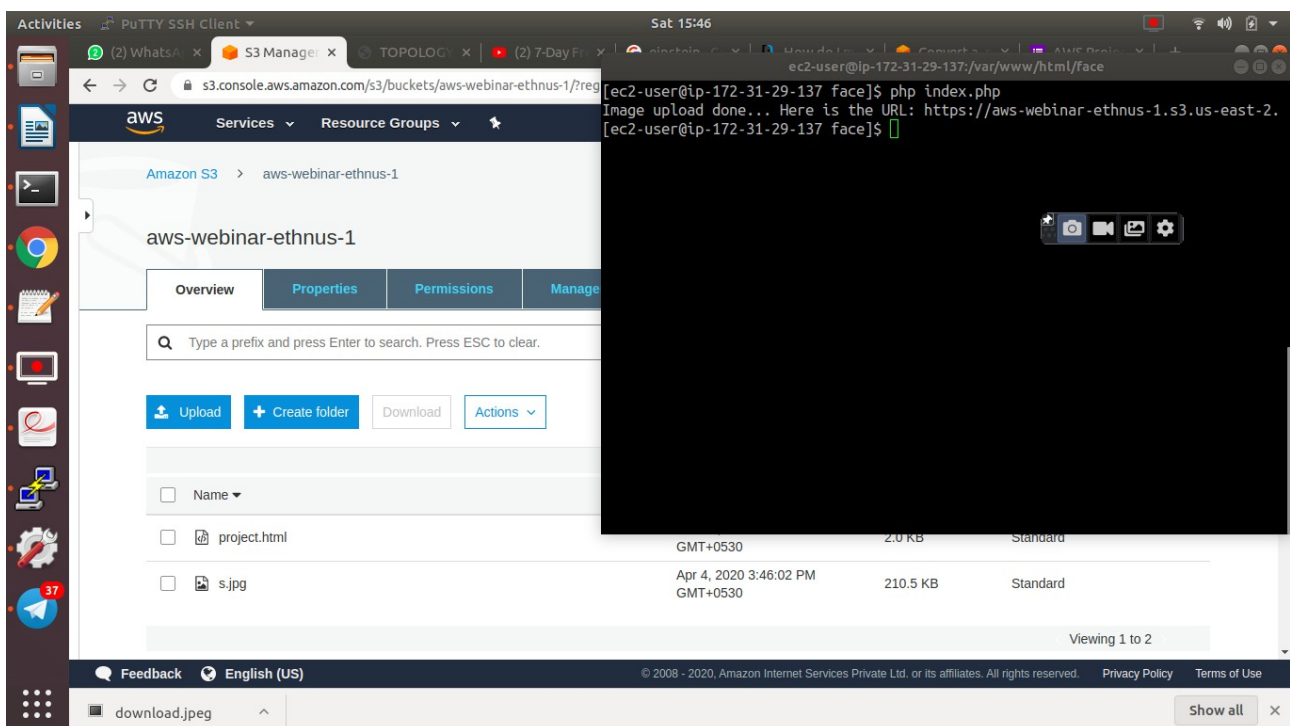
$s3 = new S3Client([
    'region' => 'us-east-2',
    'version' => '2006-03-01',
    'signature' => 'v4'
]);

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',
        ]);
    }
} catch (Exception $e) {
    echo "Error: " . $e->getMessage();
}
```

### 4. Upload success screenshot





## Screenshots needed for EC2 & Rekognition

### 1.Face Detect success screenshot

