

ETHNUS

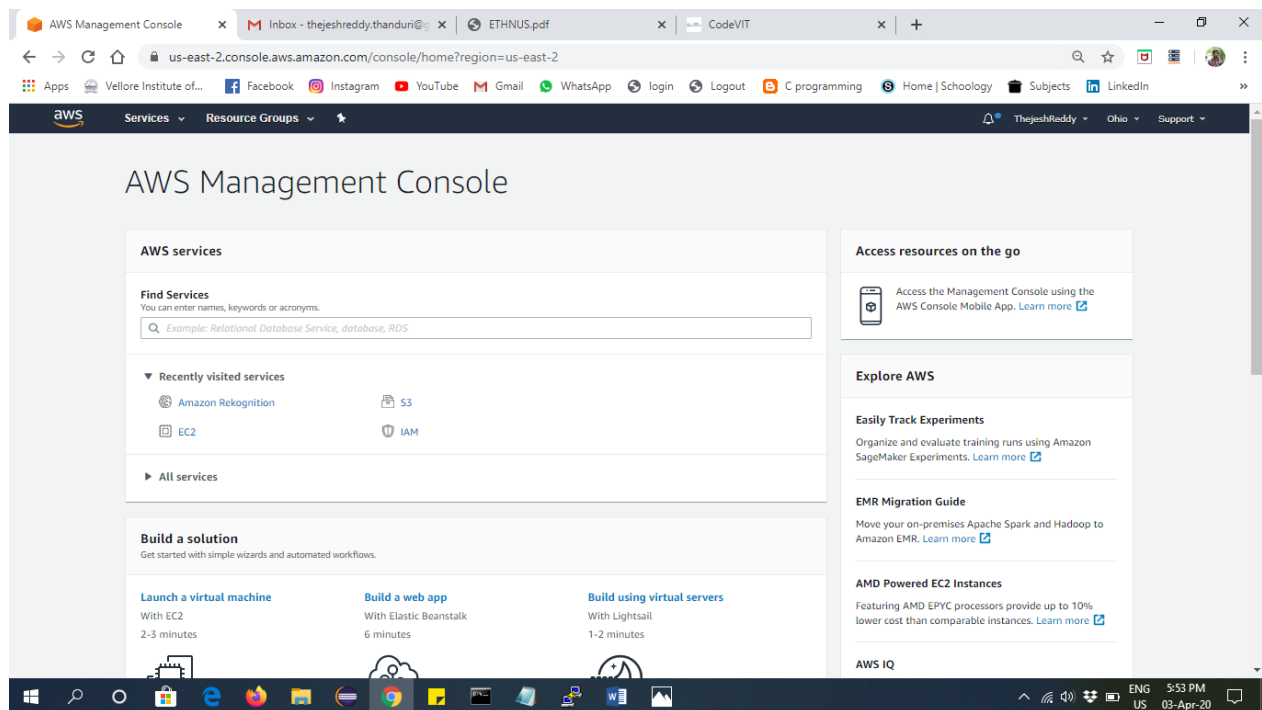
Building a Face-Detection App on AWS

NAME : T R THEJESH REDDY

REG.NO : 16MIS0246

Screenshots for Dashboards

1. AWS Login screen with username



2. EC2 Dashboard

The screenshot shows the AWS EC2 Management Console. The left sidebar contains navigation links for EC2 Dashboard, Events, Tags, Reports, Limits, INSTANCES, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, IMAGES, AMIs, Bundle Tasks, and ELASTIC BLOCK STORE. The main content area displays the 'Resources' section, showing a summary of EC2 resources in the US East (Ohio) Region. Below this, there is a 'Launch instance' section with a 'Launch instance' button and a 'Service health' section showing the status of the service. The right sidebar contains 'Account attributes' and 'Explore AWS' sections.

Resources

You are using the following Amazon EC2 resources in the US East (Ohio) Region:

| Resource | Count |
|-------------------|-------|
| Running instances | 1 |
| Elastic IPs | 0 |
| Dedicated Hosts | 0 |
| Snapshots | 0 |
| Volumes | 1 |
| Load balancers | 0 |
| Key pairs | 1 |
| Security groups | 2 |
| Placement groups | 0 |

Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

[Launch instance](#)

Note: Your instances will launch in the US East (Ohio) Region

Service health

Region: US East (Ohio) Status: ✔ This service is operating normally

Availability Zone status

| Zone | Status |
|------|--------|
|------|--------|

Account attributes

Supported platforms

- VPC

Default VPC

vpc-b3e83cd8

Console experiments

Settings

Explore AWS

Optimize your EC2 cost and performance with Spot Instances

Combine EC2 On-Demand, Spot, Savings Plans, and RIs in a single EC2 Auto Scaling Group to optimize EC2 performance and cost. [Learn more](#)

Save with AMD EPYC-Powered EC2 instances

Learn how you can use EC2 instances featuring AMD EPYC processors to deliver a 10% lower cost on compute and memory. [Read the solution brief](#)

3. S3 Dashboard

The screenshot shows the AWS S3 Management Console. The left sidebar contains navigation links for Buckets, Batch operations, Access analyzer for S3, Block public access (account settings), and Feature spotlight. The main content area displays the 'Buckets (1)' section, showing a list of buckets. The right sidebar contains a 'Create bucket' button.

Buckets (1)

[Copy ARN](#) [Empty](#) [Delete](#) [Create bucket](#)

Find bucket by name

| Name | Region | Access | Bucket created |
|------------|--------------------------|-----------------------|--------------------------|
| awsthejesh | US East (Ohio) us-east-2 | Objects can be public | 2020-04-03T07:42:20.000Z |

4. Rekognition Dashboard

The screenshot shows the Amazon Rekognition console dashboard. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', and user information 'ThejeshReddy', 'Ohio', and 'Support'. The left sidebar lists various features: 'Custom Labels' (with a 'New' tag), 'Use Custom Labels', 'Demos' (including Object and scene detection, Image moderation, Facial analysis, Celebrity recognition, Face comparison, and Text in image), 'Video Demos' (Video analysis), and 'Metrics'. The main content area features a large header with the text 'Amazon Rekognition' and 'Deep learning-based visual analysis service. Search, verify, and organize millions of images and videos'. Below this are buttons for 'Try Demo' and 'Download SDKs'. Three key benefits are highlighted: 'Easily Integrate Powerful Visual Analysis into Your App' (You don't need computer vision or deep learning), 'Continuously Learning' (Amazon Rekognition is designed to use deep learning technology to analyze billions of images), and 'Integrated with AWS Services' (Amazon Rekognition is designed to work seamlessly with other AWS services). The bottom of the dashboard includes a 'Feedback' button, 'English (US)' language selector, and copyright information '© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.' along with 'Privacy Policy' and 'Terms of Use' links. The taskbar at the bottom shows several open applications including 'index (4).php', 'index.php', 'putty-64bit-0.73-l...msi', and 'awsthejesh.pem'.

Screenshots for EC2

1. Choosing an AMI

The screenshot displays the 'Step 1: Choose an Amazon Machine Image (AMI)' screen in the AWS Launch Instance Wizard. The breadcrumb navigation at the top shows the steps: 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage, 5. Add Tags, 6. Configure Security Group, and 7. Review. The main heading is 'Step 1: Choose an Amazon Machine Image (AMI)' with a 'Cancel and Exit' link. Below the heading is a descriptive paragraph: '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.' A search bar prompts the user to 'Search for an AMI by entering a search term e.g. "Windows"'. On the left, a 'Quick Start' sidebar lists 'My AMIs', 'AWS Marketplace', and 'Community AMIs', with a 'Free tier only' filter. The main area lists three AMIs: 'Amazon Linux 2 AMI (HVM), SSD Volume Type' (ami-0e01ce4ee18447327 / ami-03201f374ab66a26e), 'Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type' (ami-01b01bbd08f24c7a8), and 'Red Hat Enterprise Linux 8 (HVM), SSD Volume Type' (ami-0520e698dd500b1d1 / ami-0099847d600887c9f). Each AMI entry includes a brief description, root device type, virtualization type, and ENA status. The 'Amazon Linux 2' and 'Amazon Linux AMI' entries show a 'Free tier eligible' badge. The 'Red Hat Enterprise Linux 8' entry also shows a 'Free tier eligible' badge. Each entry has a 'Select' button. The bottom of the screen shows the AWS footer with '© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.' and 'Privacy Policy' and 'Terms of Use' links. The taskbar at the bottom shows several open applications including 'index (4).php', 'index.php', 'putty-64bit-0.73-l...msi', and 'awsthejesh.pem'.

2. Choosing an Instance Type

Launch instance wizard | EC2 Ma | x | Inbox - thejeshreddy.thanduri@ | x | ETHNUS.pdf | x | WhatsApp | x | +

us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#LaunchInstanceWizard:

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 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 <small>Free tier eligible</small> | 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 |

Cancel Previous Review and Launch Next: Configure Instance Details

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

3. Adding Storage

Launch instance wizard | EC2 Ma | x | Inbox - thejeshreddy.thanduri@ | x | ETHNUS.pdf | x | WhatsApp | x | +

us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#LaunchInstanceWizard:

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 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.

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

4. Configuring Security Group

The screenshot shows the AWS Management Console's 'Launch instance wizard' for an EC2 instance. The user is on the '6. Configure Security Group' step. The wizard has seven steps: 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage, 5. Add Tags, 6. Configure Security Group, and 7. Review. The 'Assign a security group' section shows 'Create a new security group' selected. The 'Security group name' is 'launch-wizard-1' and the 'Description' is 'launch-wizard-1 created 2020-04-03T12:48:31.476+05:30'. Below this is a table for adding rules:

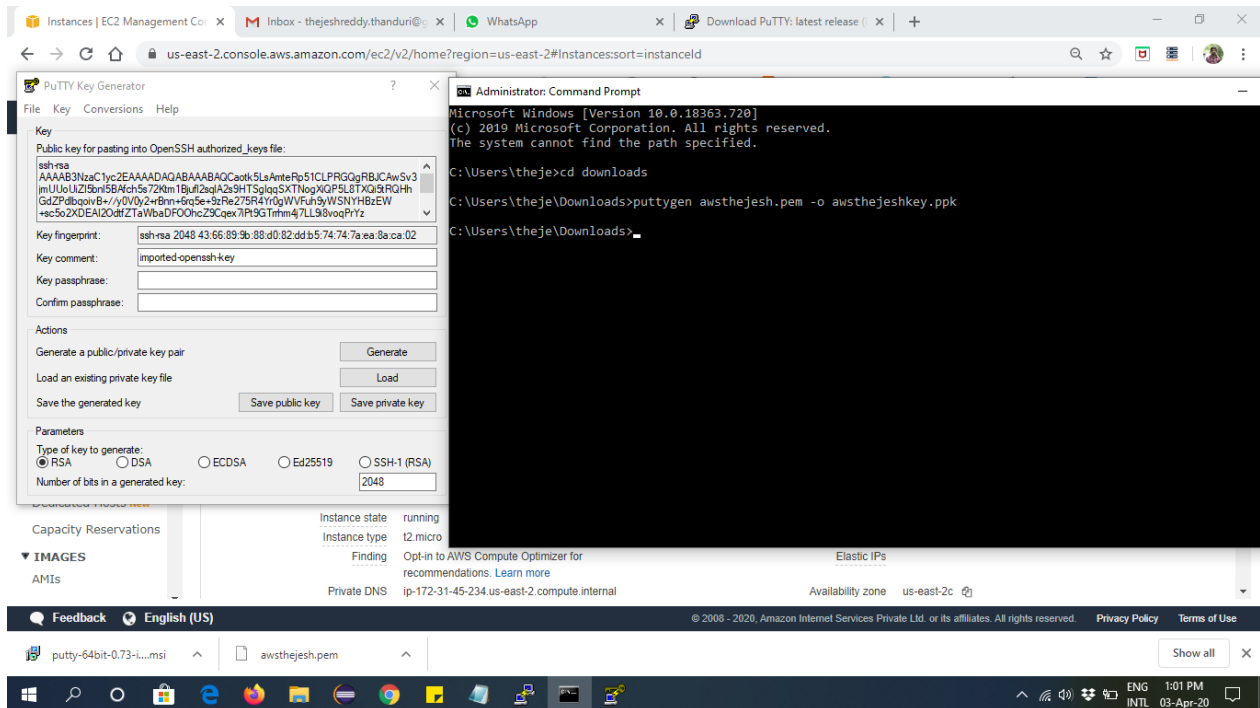
| Type | Protocol | Port Range | Source | Description |
|------|----------|------------|------------------|----------------------------|
| SSH | TCP | 22 | Custom 0.0.0.0/0 | e.g. SSH for Admin Desktop |

A warning message 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', and 'Review and Launch'.

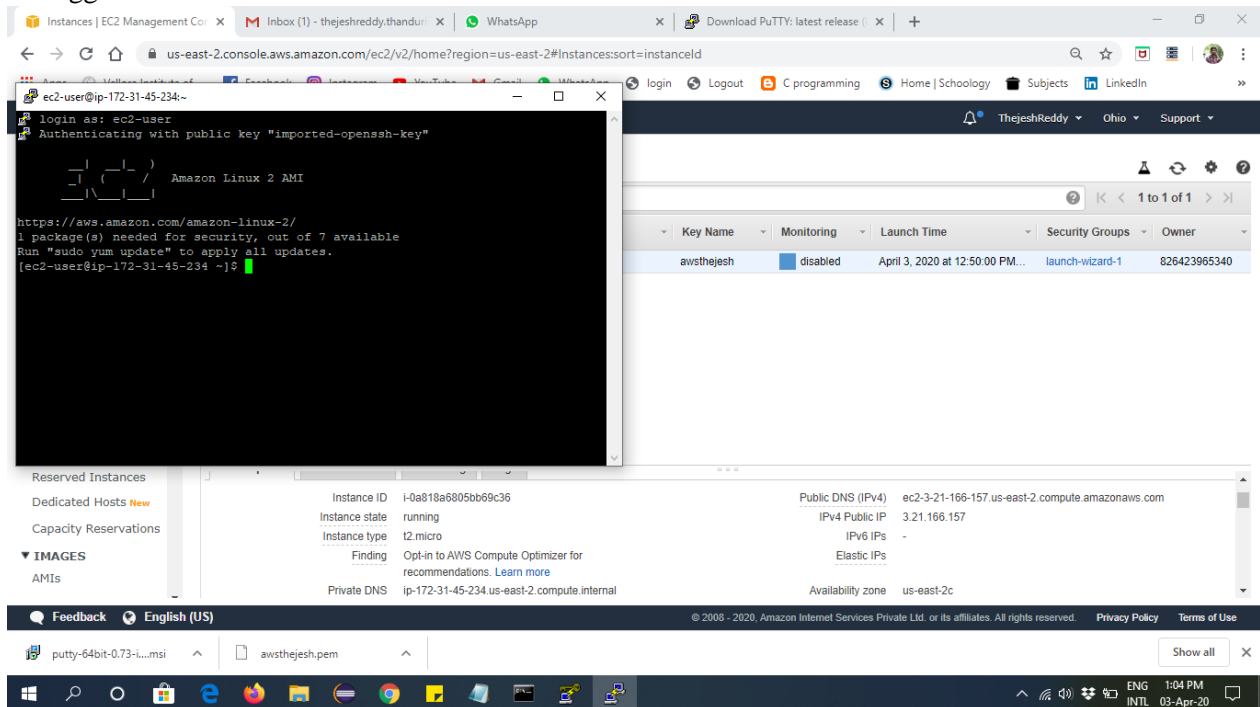
5. Key Pair Download

The screenshot shows the AWS Management Console's 'Launch instance wizard' at the '7. Review' step. A modal dialog titled 'Select an existing key pair or create a new key pair' is open. The dialog explains that a key pair consists of a public key (stored by AWS) and a private key file (stored by the user). It notes that the selected key pair will be added to the set of keys authorized for this instance. The dialog has a dropdown for 'Create a new key pair' and a text input for 'Key pair name' with the value 'awsthejesh'. A 'Download Key Pair' button is visible. A blue box with a speech bubble icon contains the instruction: '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.' At the bottom of the dialog are 'Cancel' and 'Launch Instances' buttons. In the background, the 'Review' step of the wizard is visible, showing details for the AMI (Amazon Linux 2), Instance Type (t2.micro), and Security Groups.

6. PuTTYgen conversion from pem to ppk

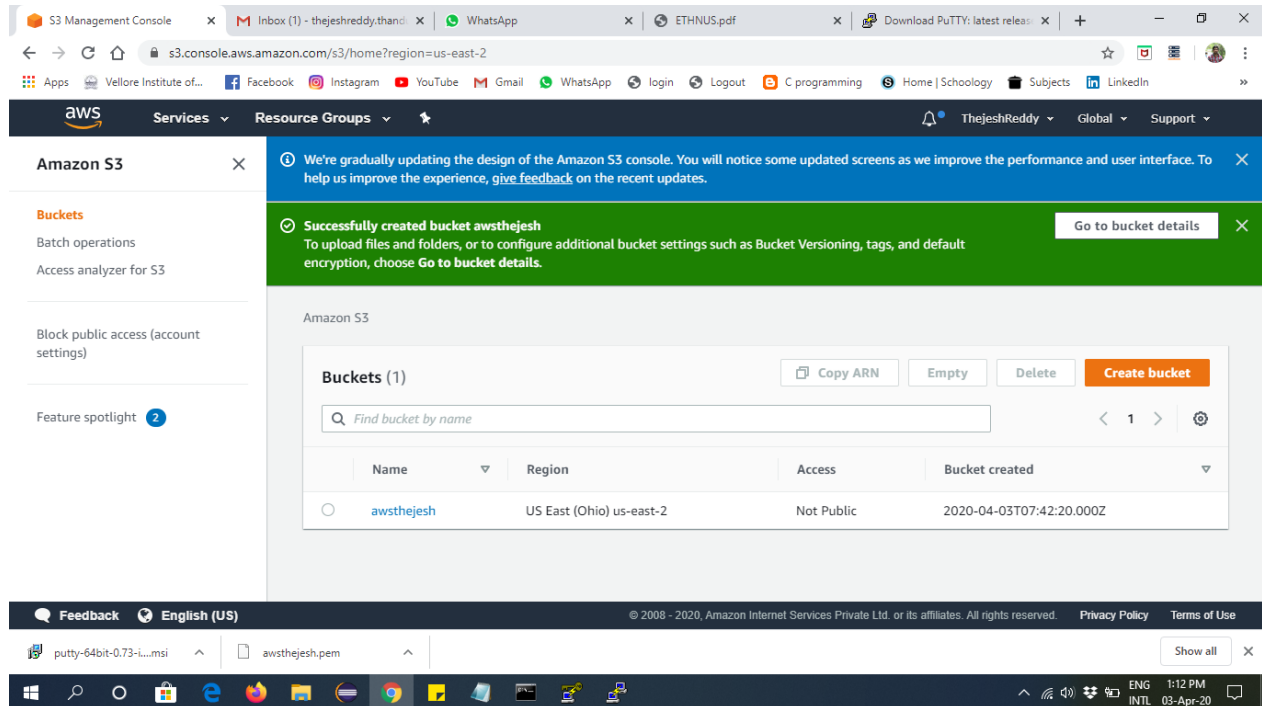


7. Logged in EC2 black screen

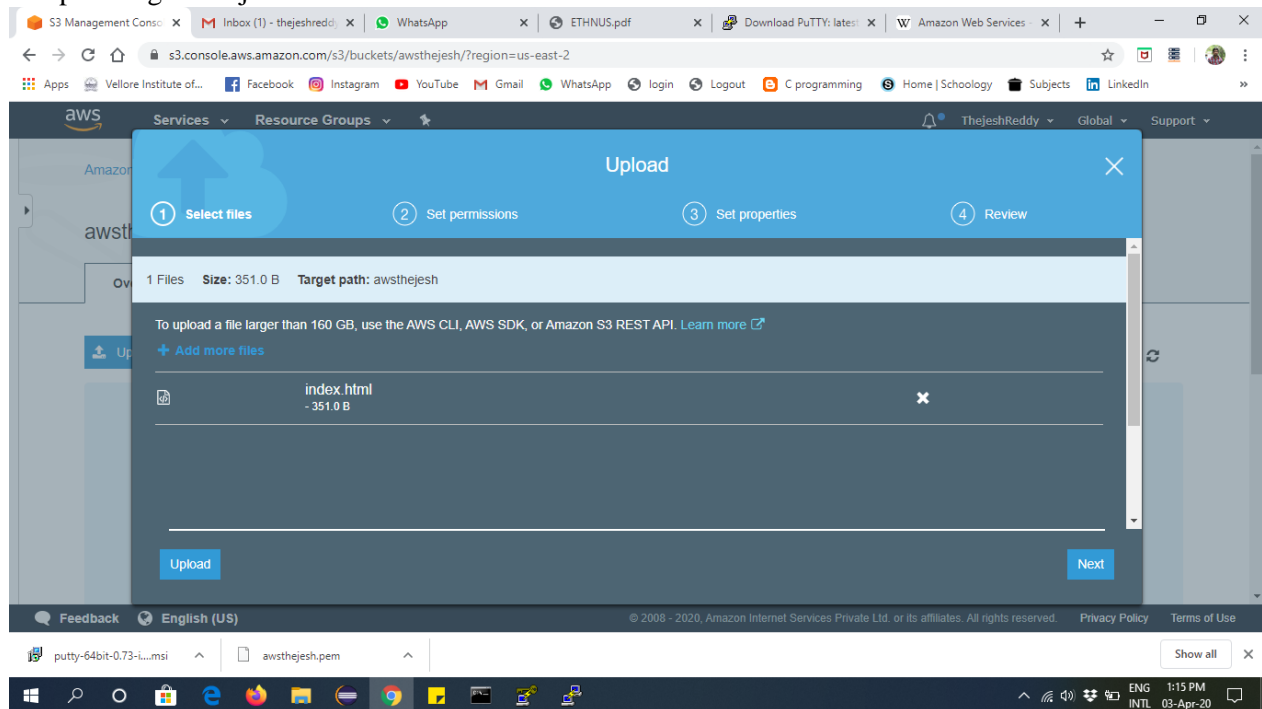


Screenshots for S3

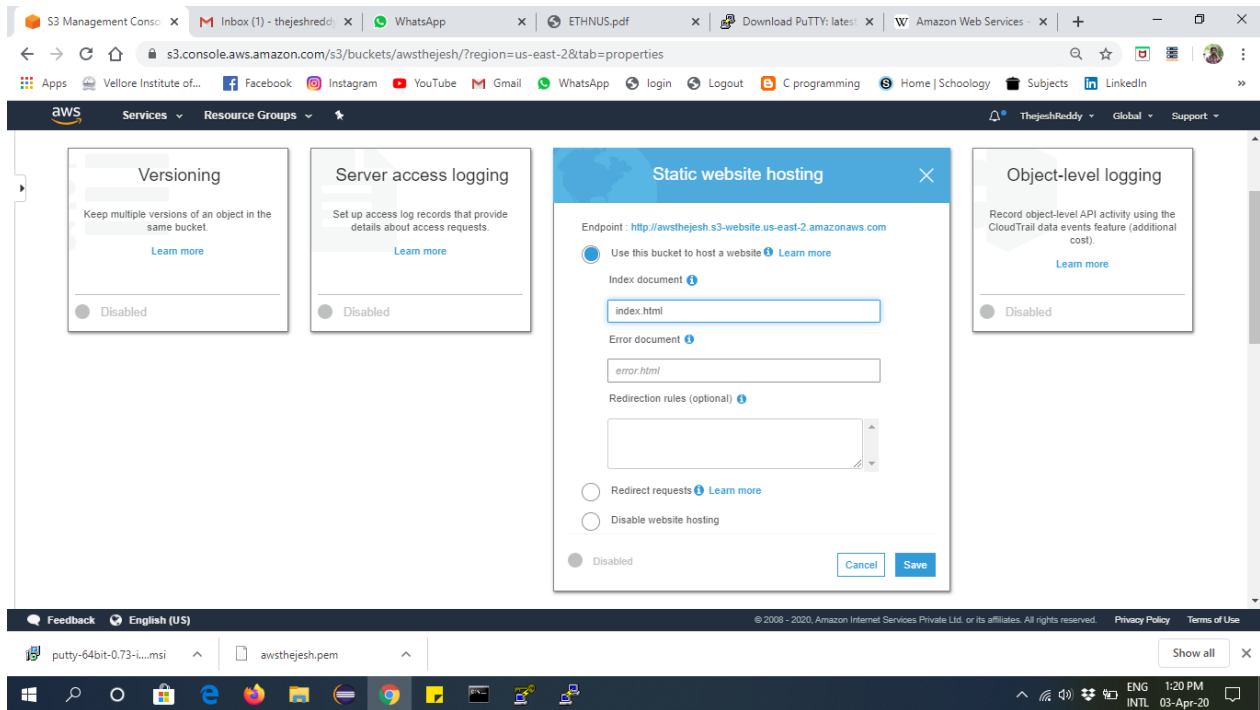
1. Creating a bucket



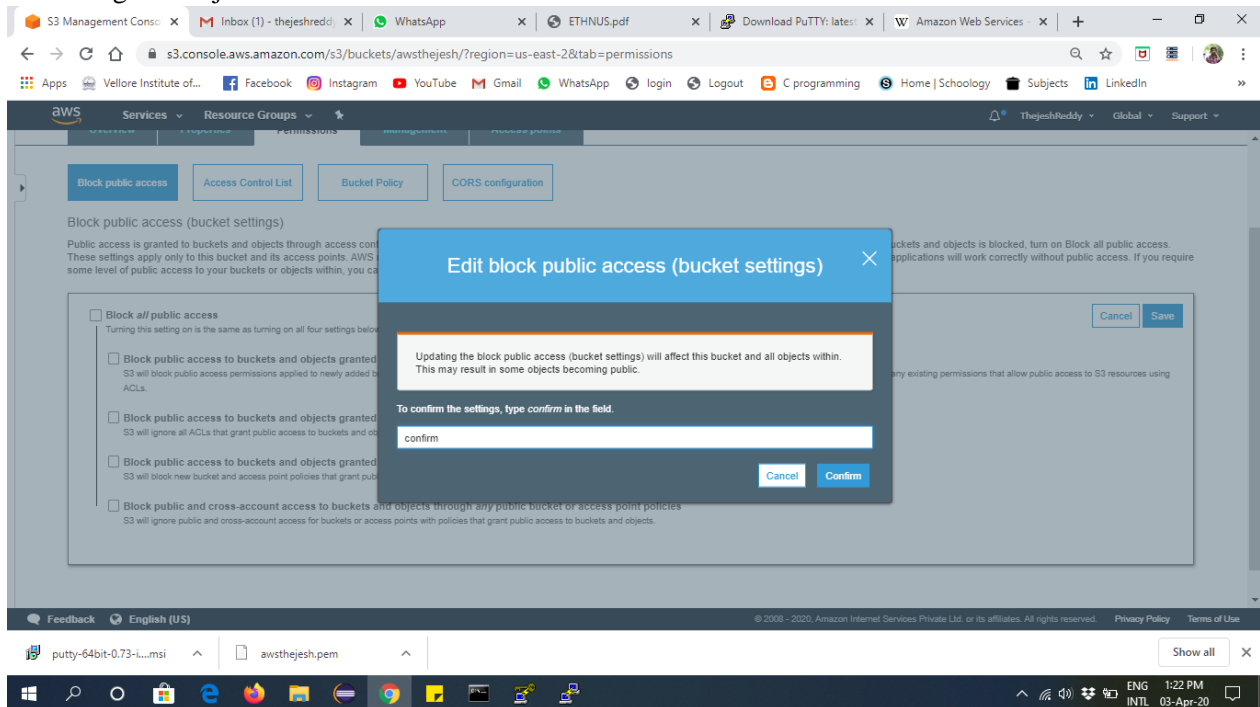
2. Uploading an Object



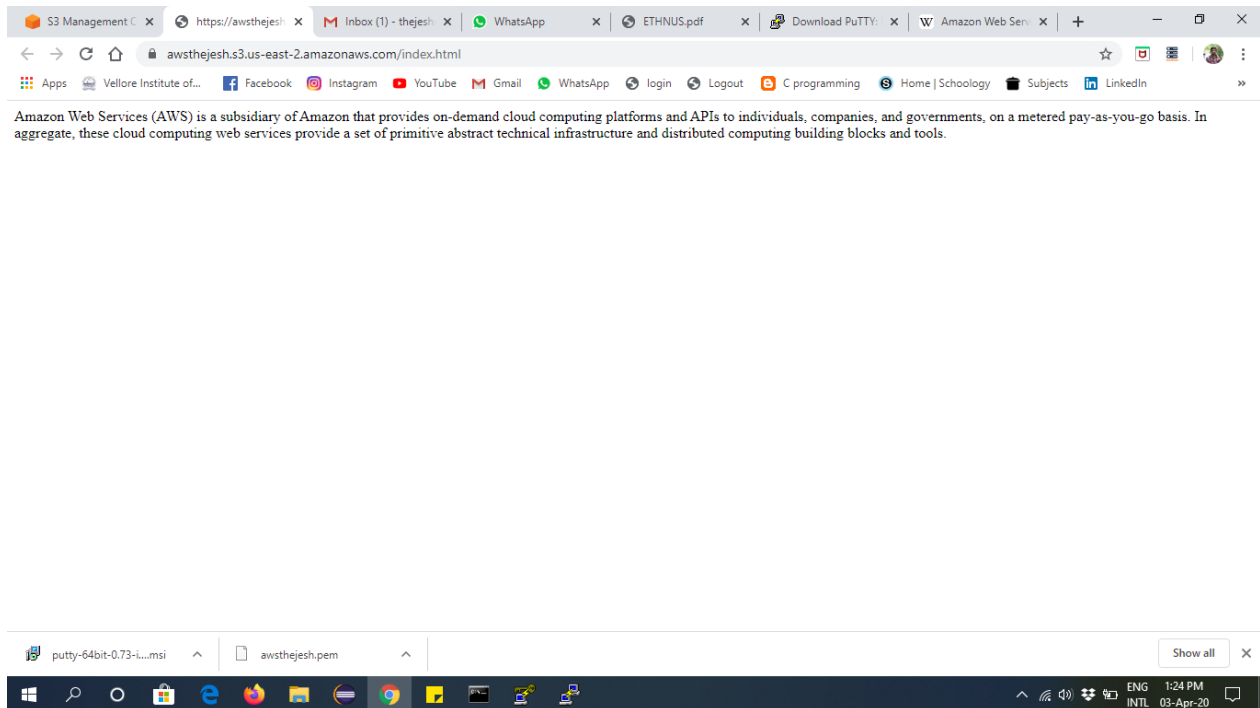
3. Enabling Static Website



4. Making the Object Public

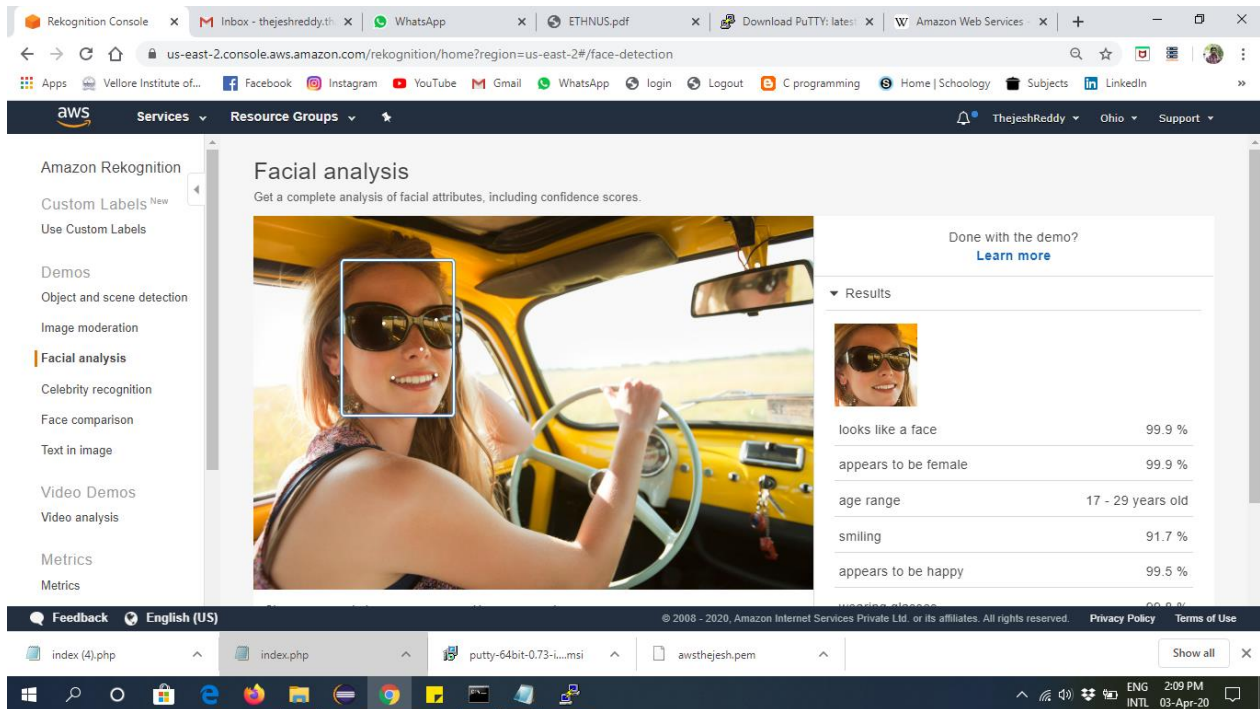


5. Checking the S3 link on the browser



Screenshots for Rekognition

1. Face Detect



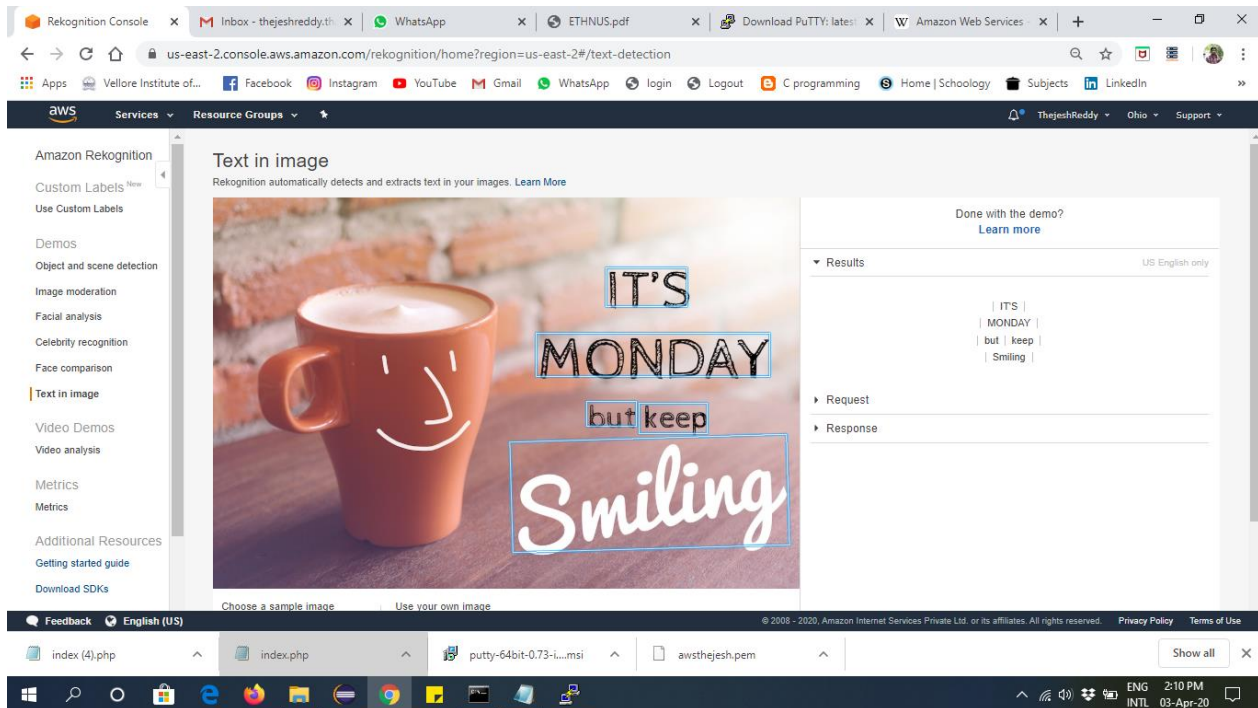
2. Face Compare

The screenshot displays the Amazon Rekognition console interface. The browser's address bar shows the URL: `us-east-2.console.aws.amazon.com/rekognition/home?region=us-east-2#/face-comparison`. The left sidebar contains a navigation menu with options like 'Custom Labels', 'Demos', 'Object and scene detection', 'Image moderation', 'Facial analysis', 'Celebrity recognition', 'Face comparison' (which is highlighted), 'Text in image', 'Video Demos', 'Video analysis', 'Metrics', and 'Additional Resources'. The main content area is titled 'Face comparison' and includes the instruction: 'Compare faces to see how closely they match based on a similarity percentage.' It features two image upload sections: 'Reference face' and 'Comparison faces'. The 'Reference face' section shows a photo of a young girl. The 'Comparison faces' section shows a photo of three young girls. Below these images are buttons to 'Choose a sample image'. On the right side, there is a 'Results' section titled 'Done with the demo? Learn more'. It displays three comparison results, each with two face images, an equals or not-equals symbol, and a similarity percentage. The first result shows a 99.8% similarity between two photos of the same girl. The other two results show lower similarity percentages. The bottom of the screen shows a Windows taskbar with various application icons and a system clock indicating 2:10 PM on 03-Apr-20.

3. Celebrity Recognition

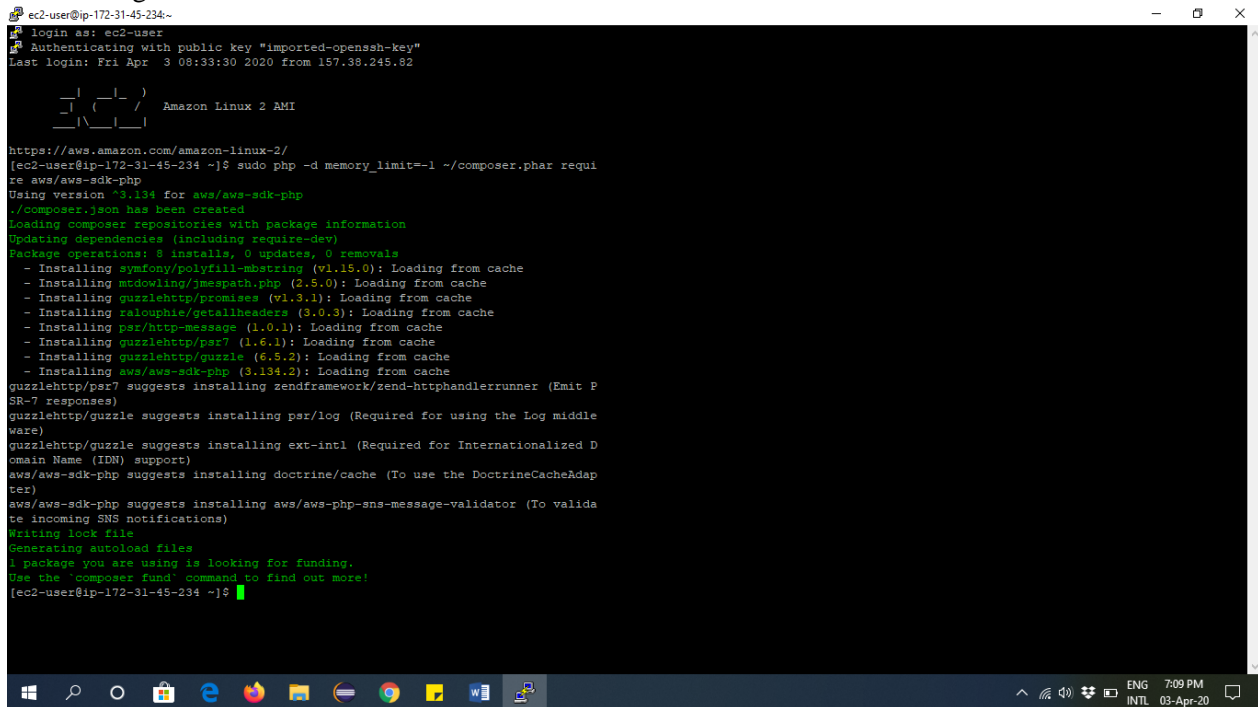
The screenshot displays the Amazon Rekognition console interface for the 'Celebrity recognition' demo. The browser's address bar shows the URL: `us-east-2.console.aws.amazon.com/rekognition/home?region=us-east-2#/celebrity-detection`. The left sidebar is similar to the previous screenshot, but 'Celebrity recognition' is highlighted. The main content area is titled 'Celebrity recognition' and includes the instruction: 'Rekognition automatically recognizes celebrities in images and provides confidence scores.' It features a large image upload section showing a photo of Jeff Bezos with a blue bounding box around his face. Below the image is a button to 'Choose a sample image'. On the right side, there is a 'Results' section titled 'Done with the demo? Learn more'. It displays the recognition result for Jeff Bezos, showing his name, a 'Learn More' link, and a 'Match confidence' of 100%. Below the result are sections for 'Request' and 'Response'. The bottom of the screen shows a Windows taskbar with various application icons and a system clock indicating 2:10 PM on 03-Apr-20.

4. Text in Image



Screenshots for EC2 & S3

1. Installing aws-sdk



Instances | EC2 Manage x | Inbox - thejeshreddy.th x | WhatsApp x | ETHNUS.pdf x

us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#instances:sort=Name

Services Resource Groups

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

Feedback English (US)

index (4).php index.php putty-64bit-0.73-l...msi

Launch Instance Connect Actions

Filter by tags and attributes or search by keyword

| Name | Instance ID | Instance Type | Availability Zone |
|------|---------------------|---------------|-------------------|
| | i-0a818a6805bb69c36 | t2.micro | us-east-2c |

Instance: i-0a818a6805bb69c36 Public DNS: ec2-3-21-166-157.us-east-2.amazonaws.com

Description Status Checks Monitoring Tags

| Instance ID | Instance state | Instance type |
|---------------------|----------------|---------------|
| i-0a818a6805bb69c36 | running | t2.micro |

```
[ec2-user@ip-172-31-45-234:~]$ sudo /bin/dd if=/dev/zero of=/var/swap.1 bs=1M count=1024
1024+0 records in
1024+0 records out
1073741824 bytes (1.1 GB) copied, 13.3832 s, 80.2 MB/s
[ec2-user@ip-172-31-45-234 face]$ sudo /sbin/mkswap /var/swap.1
mkswap: /var/swap.1: insecure permissions 0644, 0600 suggested.
Setting up swapspace version 1, size = 1024 MiB (107373728 bytes)
no label, UUID=e56f422e-56ac-4794-9alb-594f29f01849
[ec2-user@ip-172-31-45-234 face]$ sudo /sbin/swapoff /var/swap.1
swapoff: /var/swap.1: insecure permissions 0644, 0600 suggested.
[ec2-user@ip-172-31-45-234 face]$
[ec2-user@ip-172-31-45-234 face]$
[ec2-user@ip-172-31-45-234 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)
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
[ec2-user@ip-172-31-45-234 face]$
```

2. Installing php

```
ec2-user@ip-172-31-45-234:~$
- Installing guzzlehttp/promises (v1.3.1): Loading from cache
- Installing ralouphie/getallheaders (3.0.3): Loading from cache
- Installing par/http-message (1.0.1): Loading from cache
- Installing guzzlehttp/psr7 (1.6.1): Loading from cache
- Installing guzzlehttp/guzzle (6.5.2): Loading from cache
- Installing aws/aws-sdk-php (3.134.2): Loading from cache
guzzlehttp/psr7 suggests installing zendframework/zend-httphandler (Emit PSR-7 responses)
guzzlehttp/guzzle suggests installing psr/log (Required for using the Log Middleware)
guzzlehttp/guzzle suggests installing ext-intl (Required for Internationalized Domain Name (IDN) support)
aws/aws-sdk-php suggests installing doctrine/cache (To use the DoctrineCacheAdapter)
aws/aws-sdk-php suggests installing aws/aws-php-sns-message-validator (To validate incoming SNS notifications)
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-172-31-45-234 ~]$ sudo yum install php
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd, amzn2-core
Resolving Dependencies
--> Running transaction check
--> Package php.x86_64 0:7.2.28-1.amzn2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package                Arch          Version           Repository        Size
=====
Installing:
php                    x86_64        7.2.28-1.amzn2    amzn2extra-php7.2 2.9 M

Transaction Summary
-----
Install 1 Package

Total download size: 2.9 M
Installed size: 9.1 M
Is this ok [y/d/N]:
```

3. index.php file code

```
ec2-user@ip-172-31-45-234:~$ sudo /bin/dd if=/dev/zero of=/var/swap.1 bs=1M count=1024
sudo /sbin/mkswap /var/swap.1
sudo /sbin/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 = 'awshejesh';
$keyname = 'sample.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 --
```

4. success screenshot

```
ec2-user@ip-172-31-45-234:/var/www/html/face$ composer install
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.
Generating autoload files
[ec2-user@ip-172-31-45-234 face]$ sudo wget https://i.pinimg.com/originals/b9/7e/a3/b97ea33b5842c7894b804923c6c05580.jpg
--2020-04-03 08:16:31-- https://i.pinimg.com/originals/b9/7e/a3/b97ea33b5842c7894b804923c6c05580.jpg
Resolving i.pinimg.com (i.pinimg.com)... 23.210.196.239, 2600:1408:20:a91:1931, 2600:1408:20:a93:1931, ...
Connecting to i.pinimg.com (i.pinimg.com)[23.210.196.239]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 215551 (210K) [image/jpeg]
Saving to: 'b97ea33b5842c7894b804923c6c05580.jpg'

100%[=====] 215,551 --K/s in 0.04s

2020-04-03 08:16:31 (5.47 MB/s) - 'b97ea33b5842c7894b804923c6c05580.jpg' saved [215551/215551]

[ec2-user@ip-172-31-45-234 face]$ sudo mv b97ea33b5842c7894b804923c6c05580.jpg sample.jpg
[ec2-user@ip-172-31-45-234 face]$ ls
composer.json composer.lock sample.jpg vendor
[ec2-user@ip-172-31-45-234 face]$ sudo vim index.php
[ec2-user@ip-172-31-45-234 face]$ sudo php index.php
Image upload done... Here is the URL: https://awshejesh.s3.us-east-2.amazonaws.com/sample.jpg[ec2-user@ip-172-31-45-234 face]$
```


1. Face Detect success screenshot

