

***Building a Face Detection App on AWS**

using Amazon Rekognition*

- Ethnus CodeMithra 7 day MasterClass webinar
- AWS webinar : Day 4 to Day 7
- 27th March 2020 to 30th March 2020

Student Name : Swapnil Gopal Ghardale

Email : ghardaleswapnil@gmail.com

College : Nutan Maharashtra Institute of Engineering and Technology, Pune, Maharashtra, India

Department : Computer Engineering

Year : TE

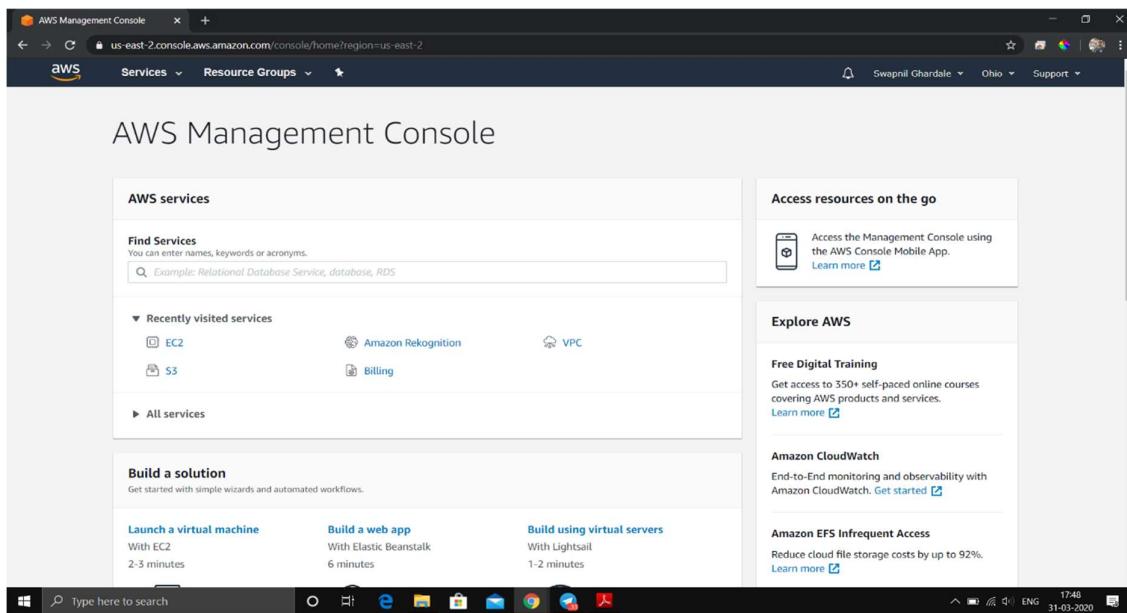
Batch : 2017-2020

Screenshots of the created project are attached below:

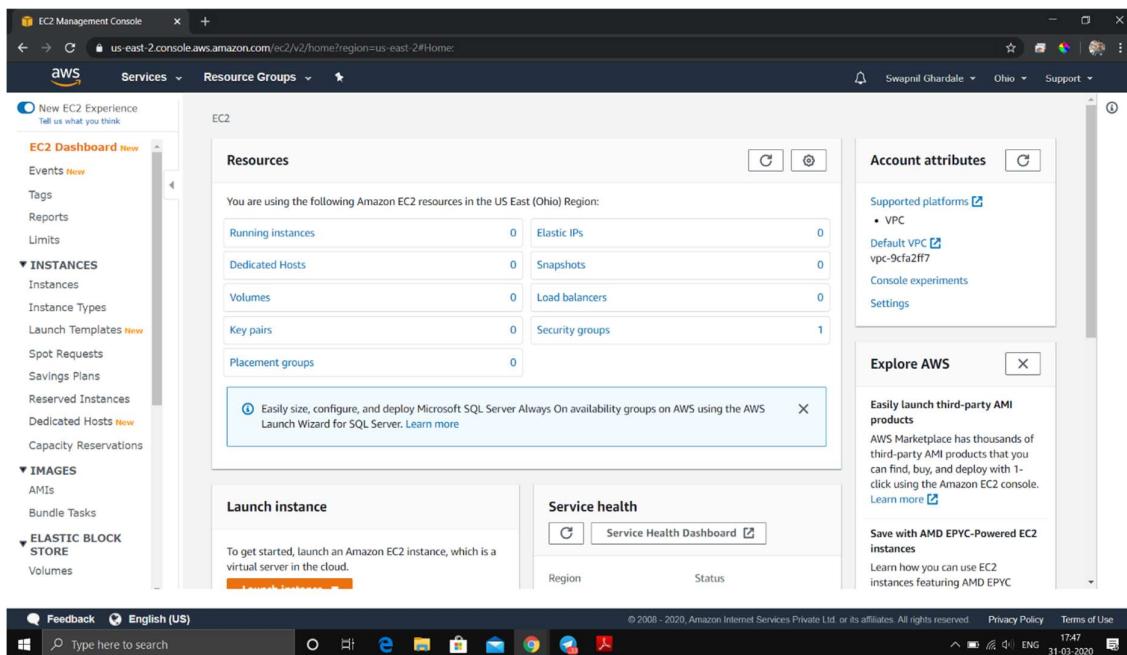
SCREENSHOTS

Screenshots for Dashboards

1. AWS Login screen with username



2. EC2 Dashboard



3. S3 Dashboard

The screenshot shows the AWS S3 Management Console interface. On the left, there's a sidebar with options like 'Buckets', 'Batch operations', 'Access analyzer for S3', and 'Block public access (account settings)'. The main area is titled 'Amazon S3' and shows a table for 'Buckets (1)'. The table has columns for 'Name', 'Region', 'Access', and 'Bucket created'. A single row is listed: 'swapnil-ghardale' (Region: US East (Ohio) us-east-2, Access: Objects can be public, Bucket created: 2020-03-29T07:10:26.000Z). At the top right of the main area, there are buttons for 'Copy ARN', 'Empty', 'Delete', and 'Create bucket'. Below the table is a search bar labeled 'Find bucket by name'. The bottom of the screen shows a Windows taskbar with various icons and a system tray indicating the date as 31-03-2020.

4. Rekognition Dashboard

The screenshot shows the AWS Rekognition Console home page. On the left, there's a sidebar with sections like 'Custom Labels', 'Demos', 'Image moderation', 'Facial analysis', 'Celebrity recognition', 'Face comparison', 'Text in image', 'Video Demos', 'Metrics', and 'Additional Resources'. The main area features a large banner with the heading 'Amazon Rekognition' and subtext 'Deep learning-based visual analysis service' and 'Search, verify, and organize millions of images and videos'. It includes a 'Try Demo' button and a 'Download SDKs' button. Below the banner, there are three sections: 'Easily Integrate Powerful Visual Analysis into Your App' (with an icon of overlapping layers), 'Continuously Learning' (with an icon of a brain and arrows), and 'Integrated with AWS Services' (with an icon of puzzle pieces). The bottom of the screen shows a Windows taskbar with various icons and a system tray indicating the date as 31-03-2020.

Screenshots for EC2

1. Choosing an AMI

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 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

Select

64-bit (x86)
 64-bit (Arm)

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

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

Select

64-bit (x86)

Red Hat Enterprise Linux 8 (HVM), SSD Volume Type - ami-0520e698dd500b1d1 (64-bit x86) / ami-0099847d600887c9f (64-bit Arm)

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

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

Select

64-bit (x86)
 64-bit (Arm)

SUSE Linux Enterprise Server 15 SP1 (HVM), SSD Volume Type - ami-04ef9b51aa142026 (64-bit x86) / ami-02a72003018010174 (64-bit Arm)

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

Type here to search

17:53 ENG 31-03-2020

2. Choosing an Instance Type

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

Family	Type	vCPUs ⓘ	Memory (GiB) ⓘ	Instance Storage (GB) ⓘ	EBS-Optimized Available ⓘ	Network Performance ⓘ	IPv6 Support ⓘ
General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
General purpose	t2.micro ⓘ Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes
General purpose	t2.xlarge	4	16	EBS only	-	Moderate	Yes
General purpose	t2.2xlarge	8	32	EBS only	-	Moderate	Yes
General purpose	t3a.nano	2	0.5	EBS only	Yes	Up to 5 Gigabit	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

Type here to search

17:54 ENG 31-03-2020

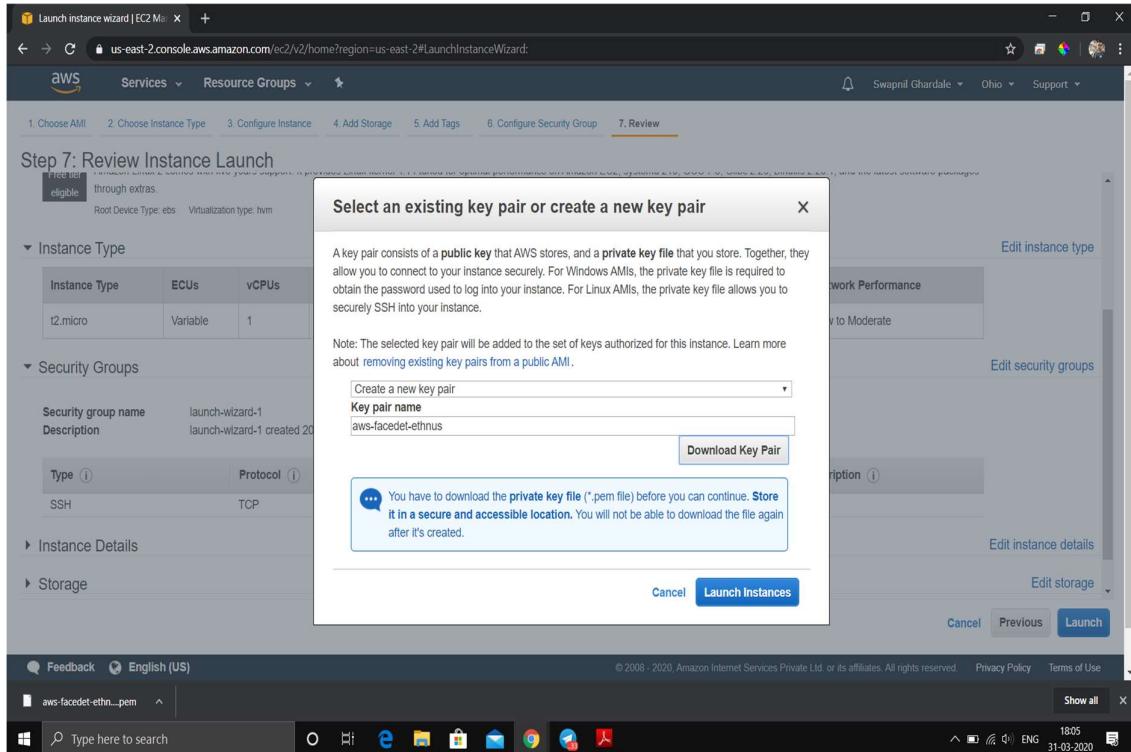
3. Adding Storage

The screenshot shows the 'Add Storage' step of the EC2 instance launch wizard. It displays a table for adding storage volumes. A single row is present for the 'Root' volume, which is mounted at '/dev/xvda'. The volume is associated with snapshot 'snap-0f54692056aaa4c20' and has a size of 8 GiB. The volume type is 'General Purpose SSD (gp2)', IOPS is 100 / 3000, and throughput is N/A. The 'Delete on Termination' checkbox is checked, and 'Encryption' is set to 'Not Encrypted'. A note below the table 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 of the page are 'Cancel', 'Previous', 'Review and Launch', and 'Next: Add Tags' buttons.

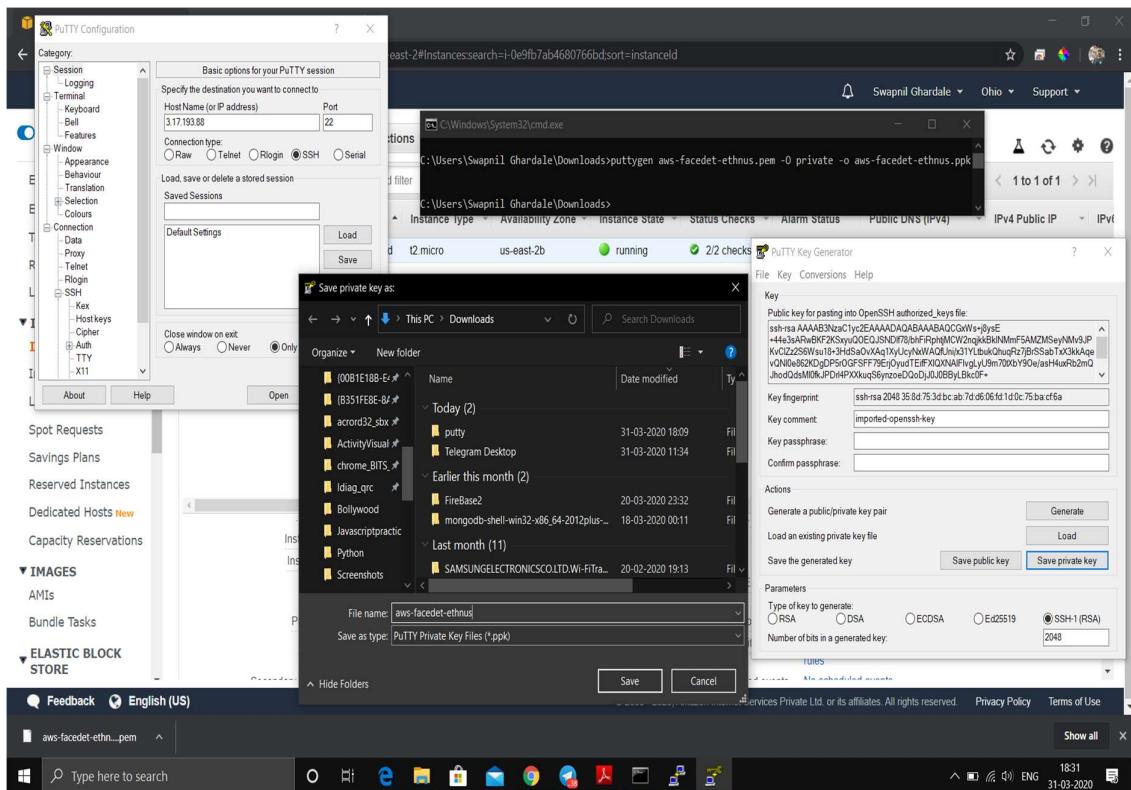
4. Configuring Security Group

The screenshot shows the 'Configure Security Group' step of the EC2 instance launch wizard. It displays a table for defining security group rules. One rule is listed: it allows SSH traffic (Protocol TCP, Port Range 22) from 'Custom' source (0.0.0.0/0) with a description 'e.g. SSH for Admin Desktop'. A warning message in a yellow box at the bottom left states: '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 of the page are 'Cancel', 'Previous', 'Review and Launch', and 'Next: Add Tags' buttons.

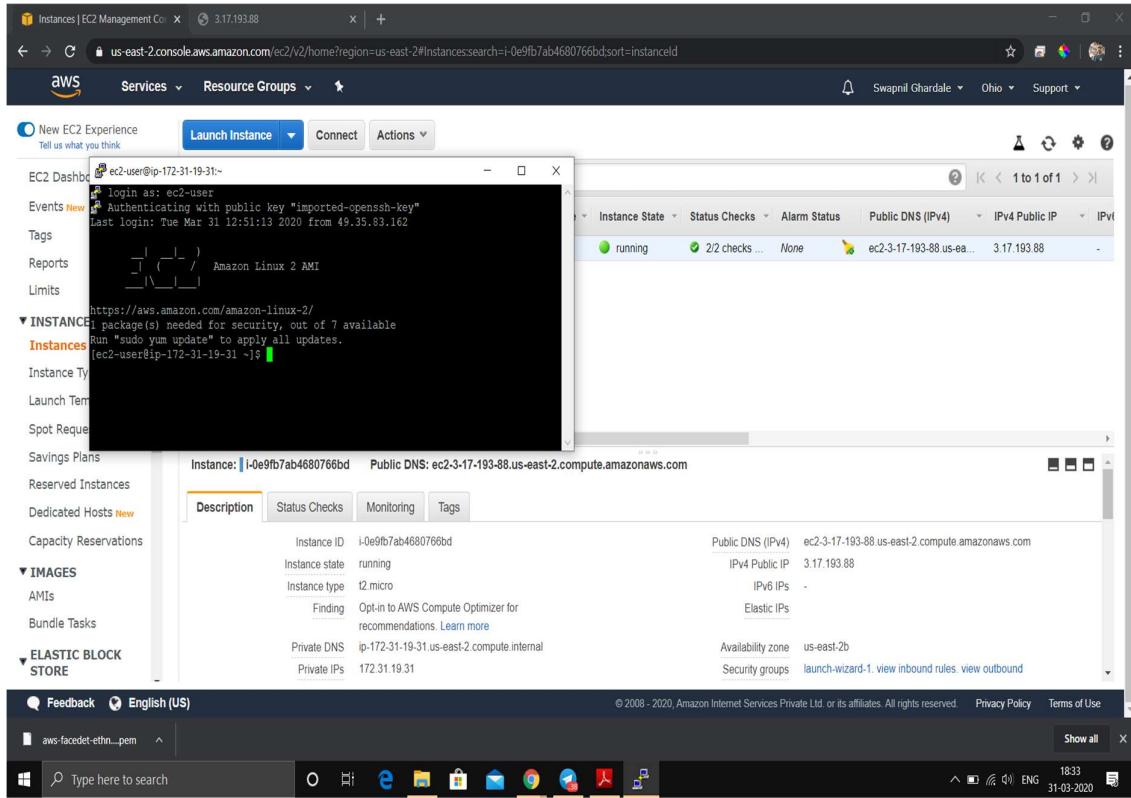
5. Key Pair Download



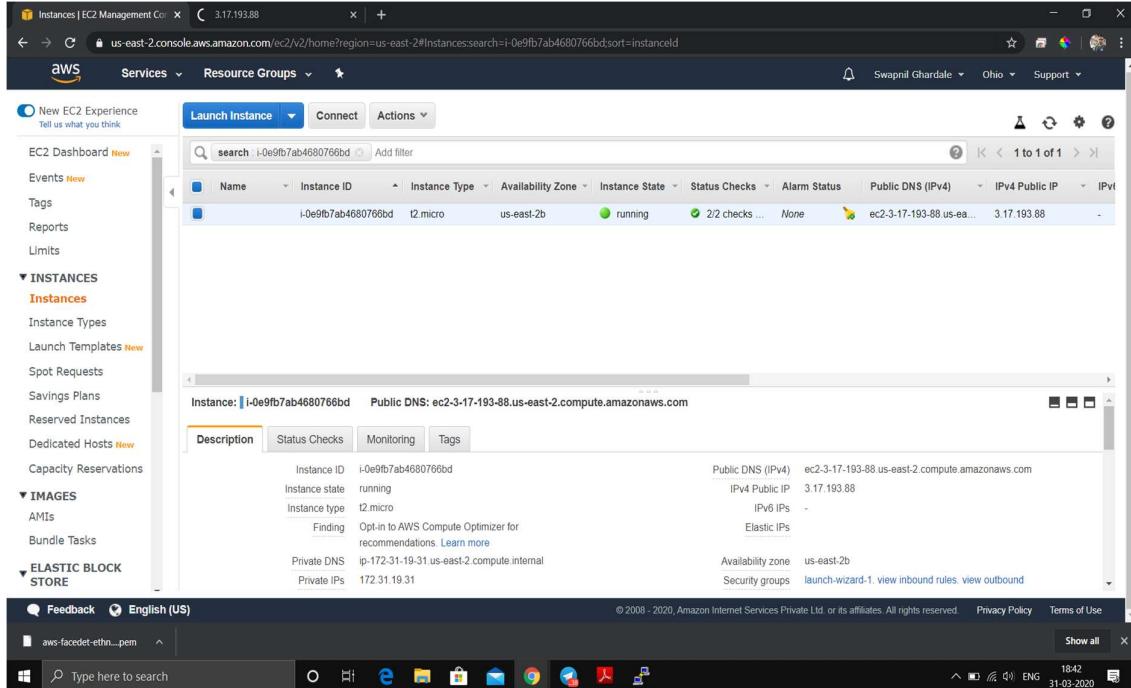
6. PuTTYgen conversion from pem to ppk



7. Logged in EC2 black screen

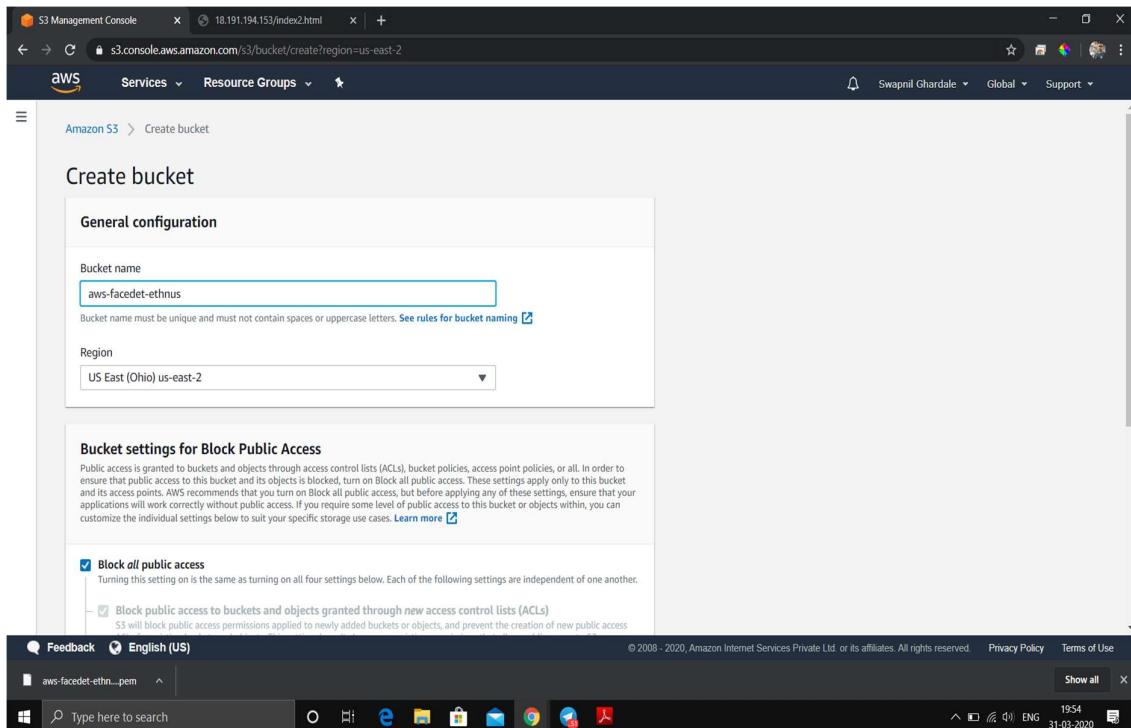


EC2 Dashboard after creating instance

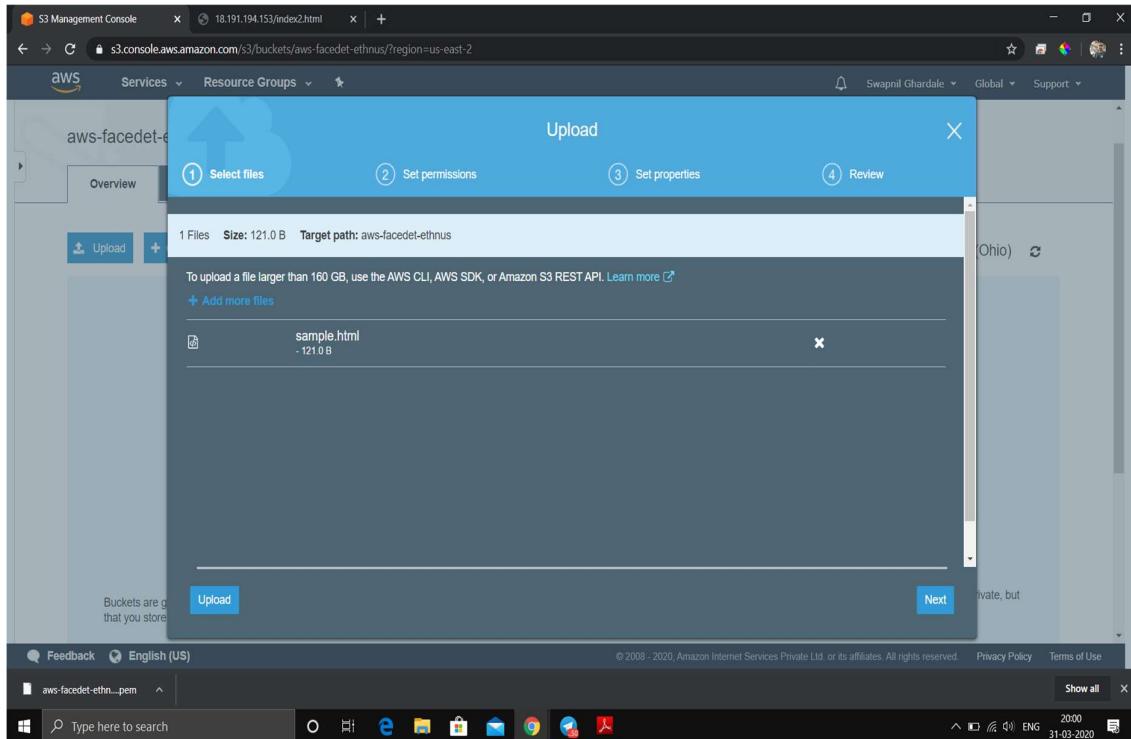


Screenshots for S3

1. Creating a bucket



2. Uploading an Object



3. Enabling Static Website

The screenshot shows the AWS S3 Management Console for the bucket 'aws-facedet-ethnus'. The 'Properties' tab is active. Under the 'Static website hosting' section, 'Bucket hosting' is enabled (indicated by a checked checkbox). Other sections like 'Versioning', 'Server access logging', and 'Object-level logging' are shown as disabled. The 'Default encryption' section indicates that objects are automatically encrypted.

Making Bucket to unblock public access

The screenshot shows the AWS S3 Management Console for the bucket 'aws-facedet-ethnus'. The 'Permissions' tab is active. Under the 'Block public access (bucket settings)' section, the 'Block all public access' setting is set to 'Off'. There are four sub-options under this setting, all also set to 'Off': 'Block public access to buckets and objects granted through new access control lists (ACLS)', 'Block public access to buckets and objects granted through any access control lists (ACLS)', 'Block public access to buckets and objects granted through new public bucket or access point policies', and 'Block public access to buckets and objects granted through any public bucket or access point policies'. The 'Edit' button is located at the top right of this section.

4. Making the Object Public

The screenshot shows the AWS S3 Management Console interface. The URL in the browser is `s3.console.aws.amazon.com/s3/object/aws-faceted-ethnus/sample.html?region=us-east-2&tab=overview`. The object name is `sample.html`. The 'Properties' tab is active. The object is set to public, as indicated by the 'Make public' button being highlighted. A modal window titled 'Make public' shows the status 'Started'. The object's details include:

- Owner:** acb0137487847b5ac85718fef1a6bc6f1e3001476a750b875e19b6fa9fbdf1e
- Last modified:** Mar 31, 2020 8:01:13 PM GMT+0530
- Etag:** 228b55948b1ea731ed0a6fdb7d26fd4
- Storage class:** Standard

The bottom of the screen shows the Windows taskbar with various pinned icons.

5. Checking the S3 link on the browser

The screenshot shows a web browser window with the URL `https://aws-faceted-ethnus.s3.us-east-2.amazonaws.com/sample.html`. The page content is:

Hello I am Swapnil Ghardale. This is my first S3 object of bucket which is public and enabled with Static website hosting

The browser taskbar shows the same pinned icons as the previous screenshot.

S3 Dashboard after creating bucket

The screenshot shows the AWS S3 Management Console interface. On the left, there's a sidebar titled "Amazon S3" with options like "Buckets", "Batch operations", "Access analyzer for S3", "Block public access (account settings)", and "Feature spotlight". The main area is titled "Amazon S3" and shows a table of "Buckets (2)". The table has columns for "Name", "Region", "Access", and "Bucket created". It lists two buckets: "aws-faceted-ethnus" (Region: US East (Ohio) us-east-2, Access: Objects can be public, Created: 2020-03-31T14:25:15.000Z) and "swapnil-ghardale" (Region: US East (Ohio) us-east-2, Access: Objects can be public, Created: 2020-03-29T07:10:26.000Z). At the top right of the table, there are buttons for "Copy ARN", "Empty", "Delete", and "Create bucket". Below the table, there's a search bar with placeholder text "Find bucket by name" and navigation arrows. The bottom of the page includes standard AWS footer links (Feedback, English (US), Privacy Policy, Terms of Use) and a Windows taskbar with various pinned icons.

Name	Region	Access	Bucket created
aws-faceted-ethnus	US East (Ohio) us-east-2	Objects can be public	2020-03-31T14:25:15.000Z
swapnil-ghardale	US East (Ohio) us-east-2	Objects can be public	2020-03-29T07:10:26.000Z

Screenshots for Rekognition

1. Face Detect / Analysis

The screenshot shows the AWS Rekognition Face Detection interface. On the left, a sidebar lists various features: Custom Labels, Use Custom Labels, Demos, Object and scene detection, Image moderation, Facial analysis (which is selected), Celebrity recognition, Face comparison, Text in image, Video Demos, Video analysis, Metrics, Additional Resources, Getting started guide, and Download SDKs. The main area is titled "Facial analysis" and contains the sub-instruction "Get a complete analysis of facial attributes, including confidence scores." Below this is a large image of a woman driving a yellow car, with a blue bounding box highlighting her face. To the right of the image is a "Results" panel displaying the following analysis:

Attribute	Value	Confidence (%)
looks like a face	99.9 %	
appears to be female	99.9 %	
age range	17 - 29 years old	
smiling	91.7 %	
appears to be happy	99.5 %	
wearing glasses	99.8 %	

Below the results are "Show more" and "Request" buttons. At the bottom of the page, there are links for Feedback, English (US), Privacy Policy, Terms of Use, and a date stamp of 31-03-2020.

2. Face Compare

The screenshot shows the AWS Rekognition Face Comparison interface. The sidebar is identical to the previous screenshot, with "Facial analysis" selected. The main area is titled "Face comparison" and contains the sub-instruction "Compare faces to see how closely they match based on a similarity percentage." It shows two images side-by-side: a "Reference face" of Amitabh Bachchan wearing glasses and a "Comparison face" of Amitabh Bachchan with a beard and different glasses. To the right is a "Results" panel showing a comparison result:

Comparison	Similarity (%)
Amitabh Bachchan (reference) vs Amitabh Bachchan (comparison)	99.8 %

Below the results are "Request" and "Response" buttons. At the bottom of the page, there are links for Feedback, English (US), Privacy Policy, Terms of Use, and a date stamp of 31-03-2020.

3. Celebrity Recognition

The screenshot shows the Amazon Rekognition console interface. On the left, a sidebar lists various features: Custom Labels, Demos (Object and scene detection, Image moderation, Facial analysis, Celebrity recognition), Metrics, Additional Resources, and Download SDKs. The 'Celebrity recognition' option is selected. The main content area is titled 'Celebrity recognition' and contains the sub-instruction 'Rekognition automatically recognizes celebrities in images and provides confidence scores.' Below this is a large image of Amitabh Bachchan wearing glasses, with a white rectangular box highlighting his face. At the bottom of this section are two buttons: 'Choose a sample image' and 'Use your own image'. To the right of the image, there's a 'Results' panel showing a thumbnail of Amitabh Bachchan with the text 'Amitabh Bachchan' and a link 'Learn More'. Below this, a section titled 'Match confidence' shows '100 %'. Underneath are sections for 'Request' and 'Response'. The top navigation bar includes tabs for 'Rekognition Console', 'https://aws-facendet-ethnus.s3.us...', '18.191.194.153/index2.html', 'covid maharashtra - Google Search', 'amitabh bachchan - Google Search', and '+'. The top right corner shows user information: Swapnil Gherdare, Ohio, Support, and a bell icon.

4. Text in Image

The screenshot shows the Amazon Rekognition console interface, similar to the previous one but with a different demo selected. The sidebar now has 'Text in image' selected under 'Demos'. The main content area is titled 'Text in image' and describes 'Rekognition automatically detects and extracts text in your images. Learn More'. It displays an image of a brown mug with a smiley face drawn on it, next to a blue sign that says 'IT'S MONDAY but keep Smiling'. Below the image are buttons for 'Choose a sample image' and 'Use your own image', with a note about file format and size. To the right is a 'Results' panel showing detected text: 'IT'S', 'MONDAY', 'but keep', and 'Smiling'. The 'Request' and 'Response' sections are also present. The top navigation bar and user information are identical to the previous screenshot.

Screenshots for EC2 & S3

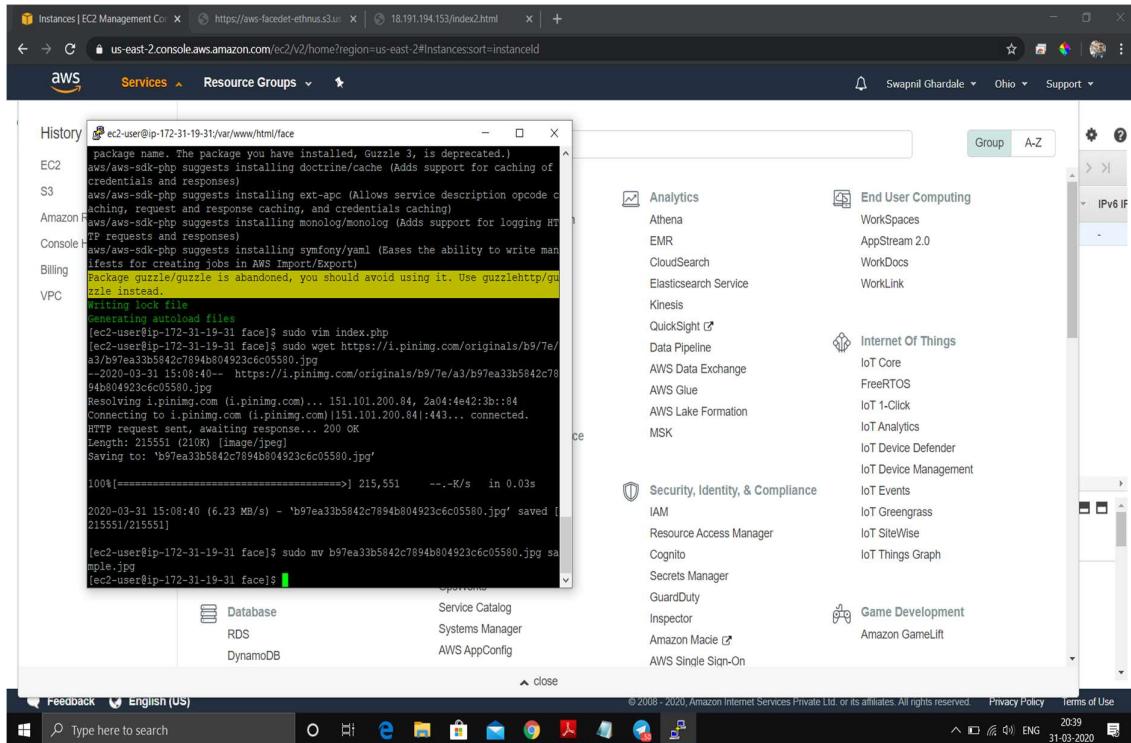
1. Installing php

```
ec2-user@ip-172-31-19-31:~$ sudo yum install php
Last login: Tue Mar 31 14:18:09 2020 from 49.35.83.162
[ec2-user@ip-172-31-19-31 ~]$ sudo yum install php
```

2. Installing aws-sdk - part1

```
Installing : libzipp010-compat-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
Verifying : php-5.4.16-46.amzn2.0.2.x86_64 4/4
Verifying : libzipp010-compat-0.10.1-9.amzn2.0.5.x86_64 1/4
Verifying : php-cl1-5.4.16-46.amzn2.0.2.x86_64 2/4
Verifying : php-common-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
Dependency Installed:
libzipp010-compat.x86_64 0:0.10.1-9.amzn2.0.5
php-cl1.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-172-31-19-31 ~]$ curl -sS https://getcomposer.org/installer | php
[ec2-user@ip-172-31-19-31 ~]$ 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-172-31-19-31 ~]$ cd /var/www/html
[ec2-user@ip-172-31-19-31 html]$ sudo mkdir face
[ec2-user@ip-172-31-19-31 html]$ cd face
[ec2-user@ip-172-31-19-31 face]$ sudo php -d memory_limit=-1 ~/composer.phar require aws/aws-sdk-php
```

Installing aws-sdk - part 2



```

History [ec2-user@ip-172-31-19-31:var/www/html/face]
package name. The package you have installed, Guzzle 3, is deprecated.)
EC2   aws/aws-sdk-php suggests installing doctrine/cache (Adds support for caching of
      credentials and responses)
S3    aws/aws-sdk-php suggests installing aop-api (Allows service description opcode c
     aching, request and response caching, and credentials caching)
Amazon RDS aws/aws-sdk-php suggests installing monolog/monolog (Adds support for logging of
      requests and responses)
Console aws/aws-sdk-php suggests installing symfony/yaml (Eases the ability to write man
      ifests for creating jobs in AWS Import/Export)
Billing package/guzzle is abandoned, you should avoid using it. Use guzzlehttp/gu
      zle instead.
VPC   writing lock file
      generating autoload files
[ec2-user@ip-172-31-19-31 face]$ sudo vim index.php
[ec2-user@ip-172-31-19-31 face]$ sudo wget https://i.pinimg.com/originals/b9/7e/a3/b97ea3b5842c7894b804923c6c05580.jpg
--2020-03-31 15:08:41-- https://i.pinimg.com/originals/b9/7e/a3/b97ea3b5842c7894b804923c6c05580.jpg
Resolving i.pinimg.com (i.pinimg.com)... 151.101.200.84, 2004:4e42:3b::84
Connecting to i.pinimg.com (i.pinimg.com)|151.101.200.84|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 215551 (21.0K) [image/jpeg]
Saving to: 'sample.jpg'

100%[=====] 215,551 --.-K/s in 0.03s

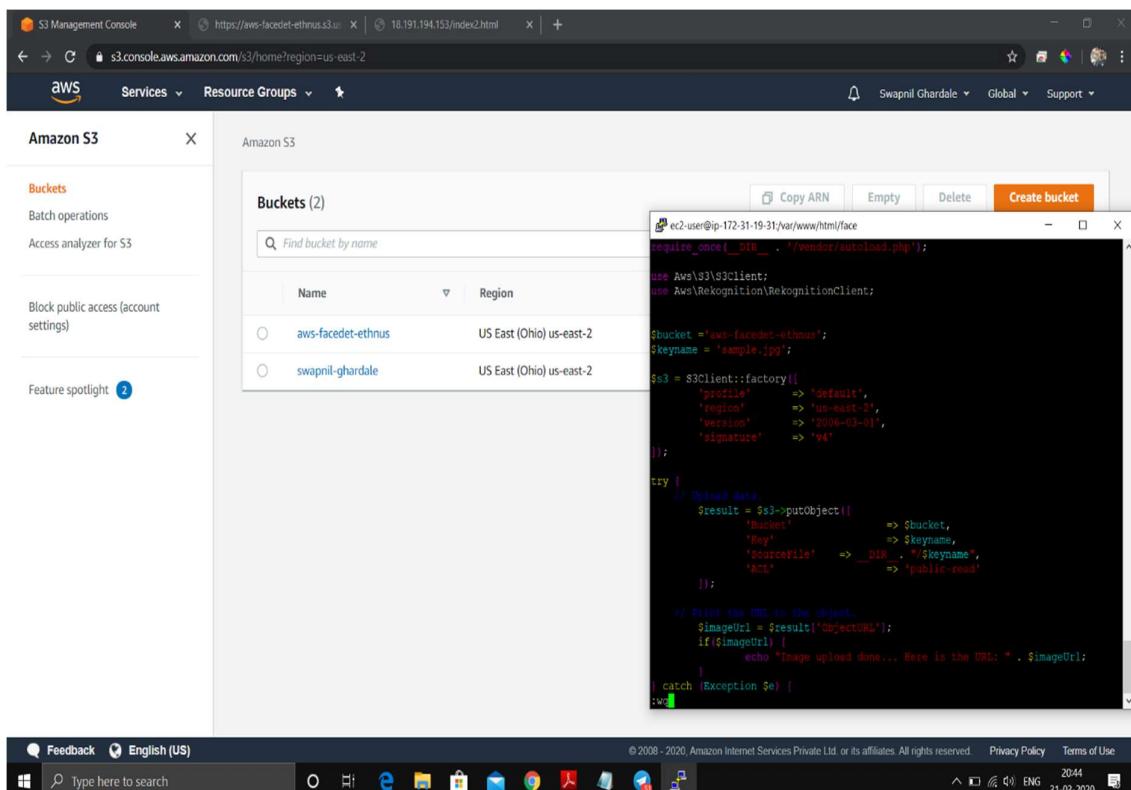
2020-03-31 15:08:40 (6.23 MB/s) - 'b97ea3b5842c7894b804923c6c05580.jpg' saved [215551/215551]

[ec2-user@ip-172-31-19-31 face]$ sudo mv b97ea3b5842c7894b804923c6c05580.jpg sample.jpg
[ec2-user@ip-172-31-19-31 face]$ 

```

The screenshot shows the AWS Management Console interface. A terminal window is open in the EC2 Instances section, displaying the command-line process of installing the aws-sdk-php package via Composer. The package is identified as deprecated, and the command `sudo wget` is used to download a sample image from i.pinimg.com. The image is successfully saved as `sample.jpg`.

3. index.php file code



```

Buckets
Batch operations
Access analyzer for S3

Block public access (account settings)

Feature spotlight 2

Buckets (2)
Name Region
aws-faceted-ethnus US East (Ohio) us-east-2
swapnil-ghardale US East (Ohio) us-east-2

Copy ARN Empty Delete Create bucket
$ ec2-user@ip-172-31-19-31:var/www/html/face
require_once(__DIR__ . '/vendor/autoload.php');

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

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

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

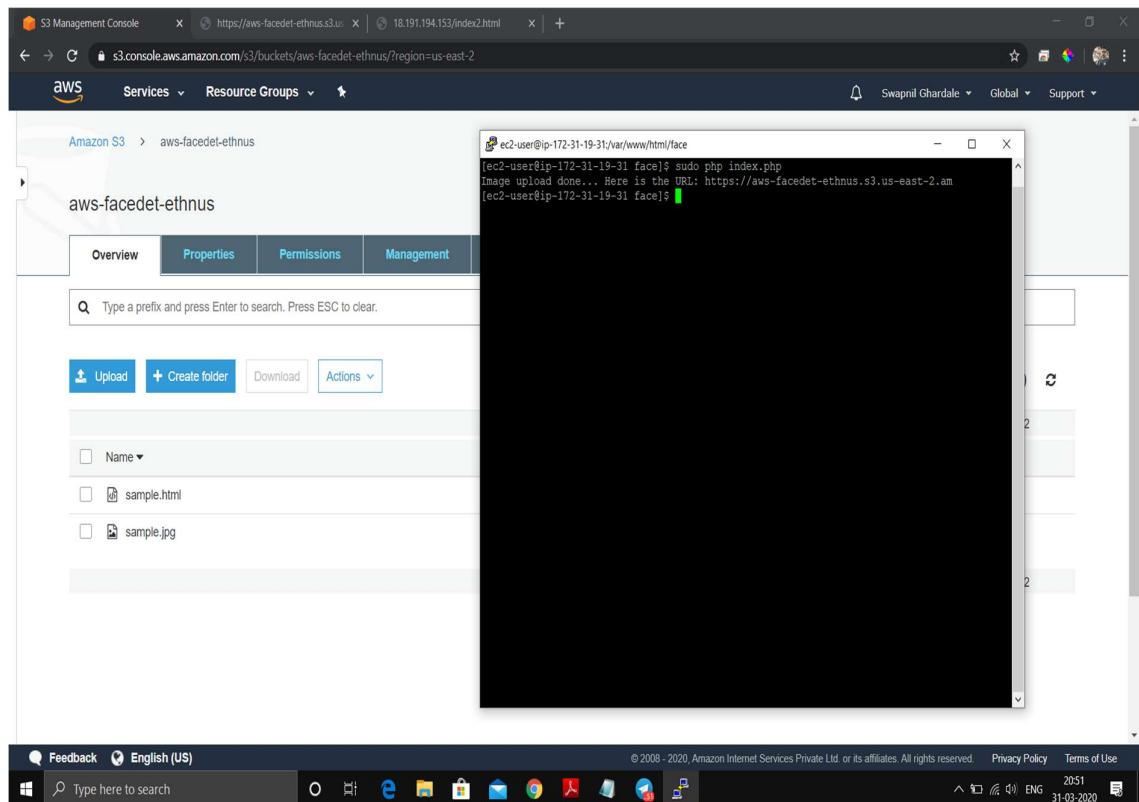
try {
    // Upload data.
    $result = $s3->putObject([
        'Bucket'      => $bucket,
        'Key'         => $keyname,
        'SourceFile'  => __DIR__ . '/' . $keyname,
        'ACL'         => 'public-read'
    ]);

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

```

The screenshot shows the AWS Management Console interface. A terminal window is open in the S3 Buckets section, displaying the command-line process of uploading a file named `sample.jpg` to an S3 bucket named `aws-faceted-ethnus`. The file is uploaded with a public-read ACL. The code also includes logic to print the generated URL.

4. Upload success screenshot



*NOTE: As we stop and restart the ec2 instance the public DNS and public IPv4 Address gets changed, keeping the private address same for particular instance.

In this case, I have stopped and restarted the instance which changed instances' public IPv4 address but the private IP address remains same.

Screenshots for EC2 & Rekognition

1. Face Detect success screenshot

The screenshot shows the AWS S3 Management Console and a terminal window on an EC2 instance.

In the S3 Management Console:

- The top navigation bar shows "S3 Management Console", "https://aws-faceted-ethnus.s3.us-east-2.amazonaws.com", "18.191.194.153/face/index.php", and a "+" button.
- The main area displays two files: "sample.html" and "sample.jpg".
- File details:
 - "sample.html": Last modified Mar 31, 2020 8:01:13 PM GMT+0530, Size 121.0 B, Storage class Standard.
 - "sample.jpg": Last modified Mar 31, 2020 11:05:41 PM GMT+0530, Size 210.5 KB, Storage class Standard.

In the terminal window:

```
ec2-user@ip-172-31-19-31:~$ sudo vim index.php
[ec2-user@ip-172-31-19-31:~$] sudo php index.php
image upload done... Here is the URL: https://aws-faceted-ethnus.s3.us-east-2.amazonaws.com/sample.jpg totally there are 9 faces[ec2-user@ip-172-31-19-31:~$]
```

Updated index.php with Rekognition function invoked.

The screenshot shows the AWS S3 Management Console and a terminal window on an EC2 instance, demonstrating the updated index.php code.

In the S3 Management Console:

- The top navigation bar shows "S3 Management Console", "https://aws-faceted-ethnus.s3.us-east-2.amazonaws.com", "18.191.194.153/index2.html", and a "+" button.
- The main area displays one file: "sample.html".
- File details:
 - "sample.html": Last modified Mar 31, 2020 11:05:41 PM GMT+0530, Size 210.5 KB, Storage class Standard.

In the terminal window:

```
ec2-user@ip-172-31-19-31:~$ cat index2.html
<html>
<head>
<title>Amazon S3 - aws-faceted-ethnus</title>
</head>
<body>
    <h1>aws-faceted-ethnus</h1>
    <ul>
        <li><a href="#">Overview</a></li>
        <li><a href="#">Properties</a></li>
        <li><a href="#">Permissions</a></li>
        <li><a href="#">Management</a></li>
        <li><a href="#">Access points</a></li>
    </ul>
    <pre>
$bucket = 'aws-faceted-ethnus';
$keyname = 'sample.jpg';

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

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

    // Get 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',
    'signature'   => 'v4'
]);

$result = $rekognition->detectFaces([
    'AttributeList' => ['SUBTITLE'],
    'Image'        => [
        'S3Object' => [
            'Bucket'  => $bucket,
            'Name'    => $keyname,
        ]
    ]
]);
-- INSERT --
    </pre>
</body>
</html>
```