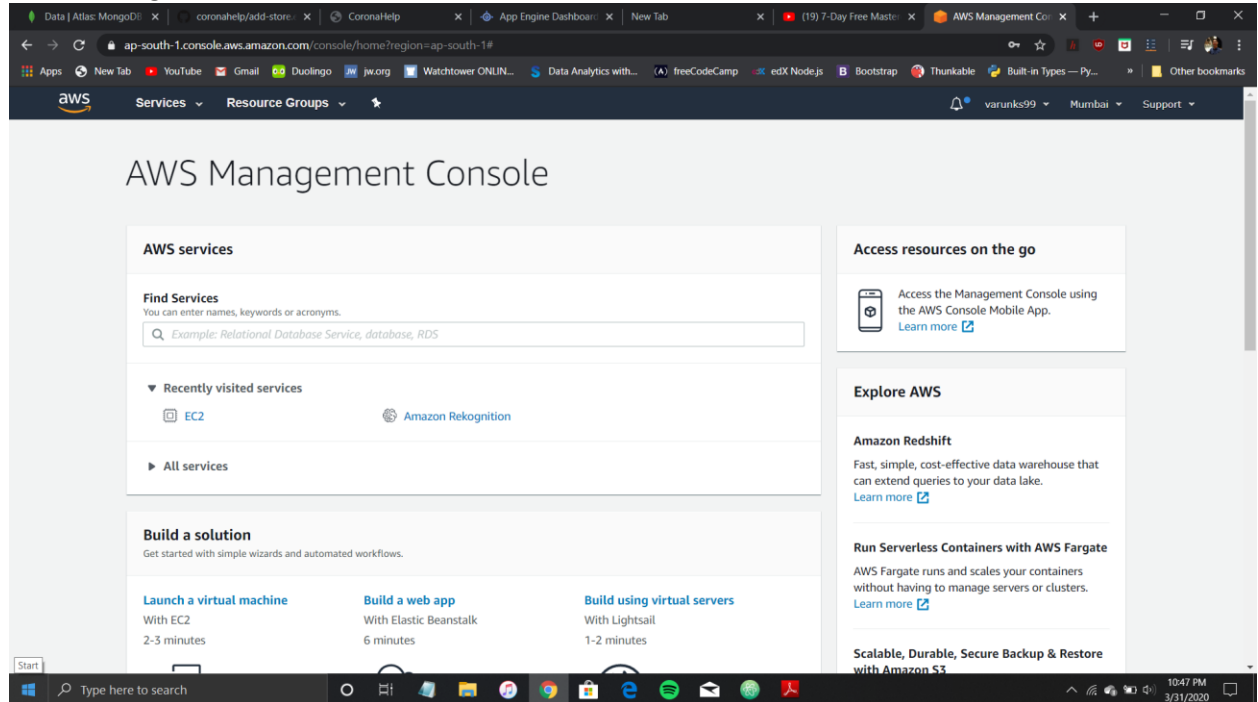


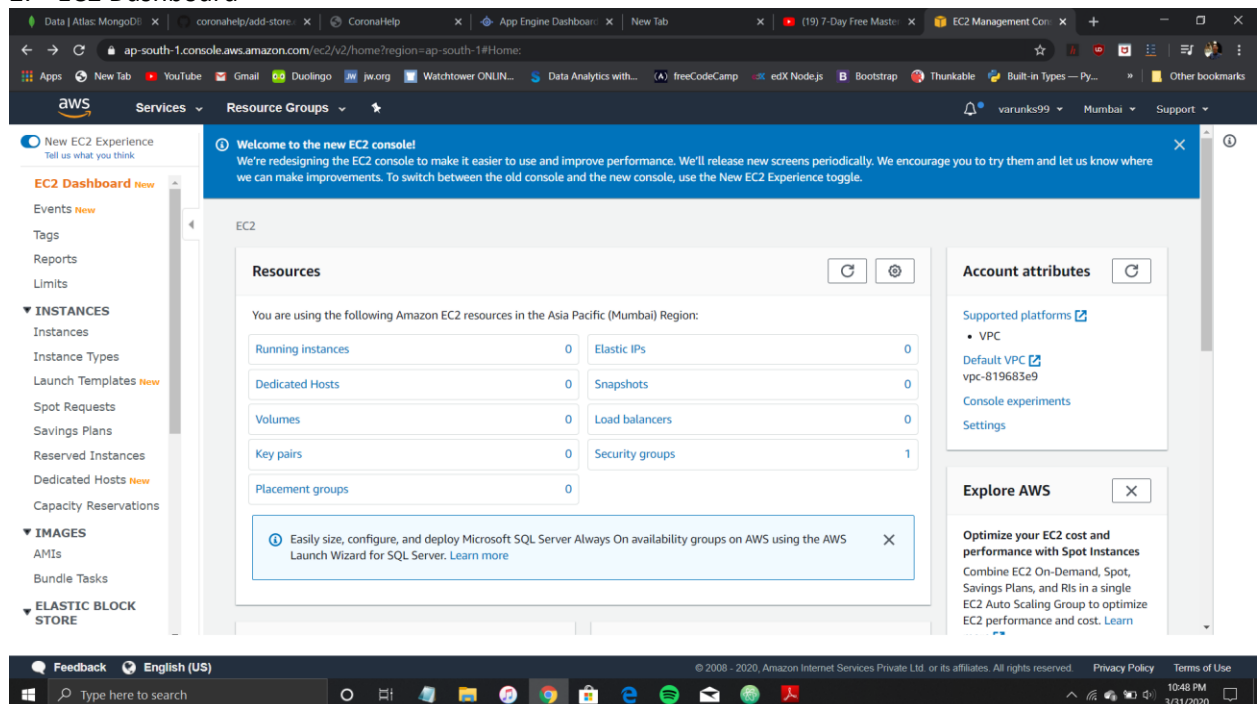
AWS Face Recognition Project

Name: Varun Shiri

1. AWS Login screen with username



2. EC2 Dashboard



3. S3 Dashboard

The screenshot shows the Amazon S3 console in the 'ap-south-1' region. A notification banner at the top states: 'We're gradually updating the design of the Amazon S3 console. You will notice some updated screens as we improve the performance and user interface. To help us improve the experience, give feedback on the recent updates.' The left sidebar contains navigation links: Buckets, Batch operations, Access analyzer for S3, Block public access (account settings), and Feature spotlight. The main content area, titled 'Amazon S3', shows 'Buckets (1)' with a search bar and a table of buckets.

Name	Region	Access	Bucket created
face-recognition-vks99	Asia Pacific (Mumbai) ap-south-1	Objects can be public	2020-04-01T11:08:51.000Z

At the bottom of the console, there is a footer with copyright information: '© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.' and links for Privacy Policy and Terms of Use. The system clock shows 3:32 PM on 4/5/2020.

4. Rekognition Dashboard

The screenshot displays the Amazon Rekognition dashboard. The left sidebar lists various features: Demos (Object and scene detection, Image moderation, Facial analysis, Celebrity recognition, Face comparison, Text in image), Video Demos (Video analysis), Metrics, and Additional Resources (Getting started guide, Download SDKs, Developer resources). The main content area features a large hero section with the title 'Amazon Rekognition' and the subtitle 'Deep learning-based visual analysis service'. It includes a 'Try Demo' button and a 'Download SDKs' link. Below the hero section, there are three columns of information:

- Easily Integrate Powerful Visual Analysis into Your App**: You don't need computer vision or deep learning expertise to take advantage of Rekognition's high quality image and video analysis for your web, mobile, enterprise or device applications. Amazon Rekognition removes the complexity of building...
- Continuously Learning**: Amazon Rekognition is designed to use deep learning technology to analyze billions of images and videos daily. It is continuously learning as we add support for new capabilities and learn from more and more data.
- Integrated with AWS Services**: Amazon Rekognition is designed to work seamlessly with other AWS services. Rekognition integrates directly with Amazon S3 and AWS Lambda so you can build scalable, affordable, and reliable visual analysis applications. You can start analyzing images and videos stored in Amazon S3...

The footer of the dashboard includes the same copyright notice and links as the S3 dashboard, along with the system clock showing 10:31 PM on 4/4/2020.

EC2

1. Choosing an AMI

The screenshot shows the AWS Management Console's 'Launch Instance Wizard' at the 'Choose an Amazon Machine Image (AMI)' step. The breadcrumb trail indicates the sequence: 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage, 5. Add Tags, 6. Configure Security Group, 7. Review. The page title is 'Step 1: Choose an Amazon Machine Image (AMI)'. Below the title, it explains that an AMI is a template containing the operating system, application server, and applications. A search bar is provided with the placeholder text 'Search for an AMI by entering a search term e.g. "Windows"'. The 'Quick Start' section on the left lists 'My AMIs', 'AWS Marketplace', and 'Community AMIs', with a 'Free tier only' filter selected. The main content area displays 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 entry includes its ID, a brief description, and a 'Select' button. The 'Amazon Linux 2' AMI is highlighted as the selected option.

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

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-03b5297d565ef30a6 (64-bit x86) / ami-0292503f80fe49021 (64-bit Arm)

Amazon Linux 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-0fa6cd5aefbf02afe

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-0a74f6feb190bd404f

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

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

2. Choosing an Instance Type

The screenshot shows the AWS Management Console's 'Launch Instance Wizard' at the 'Choose an Instance Type' step. The breadcrumb trail indicates the sequence: 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage, 5. Add Tags, 6. Configure Security Group, 7. Review. The page title is 'Step 2: Choose an Instance Type'. Below the title, it explains that Amazon EC2 provides a wide selection of instance types optimized for different use cases. A 'Filter by' section allows filtering by 'All instance types', 'Current generation', and 'Show/Hide Columns'. The 'Currently selected' section shows 't2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)'. A table lists various instance types with columns for Family, Type, vCPUs, Memory (GiB), Instance Storage (GB), EBS-Optimized Available, Network Performance, and IPv6 Support. The 't2.micro' instance type is highlighted as the selected option. At the bottom, there are buttons for 'Cancel', 'Previous', 'Review and Launch', and 'Next: Configure Instance Details'.

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	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

3. Adding Storage

The screenshot shows the AWS Management Console at the URL `ap-south-1.console.aws.amazon.com/ec2/v2/home?region=ap-south-1#LaunchInstanceWizard`. The navigation bar at the top includes the AWS logo, 'Services', 'Resource Groups', and a user profile 'varunks99' from 'Mumbai'. The wizard progress bar shows steps 1 through 7, with '4. Add Storage' currently selected.

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-0a65864f0ac2b05f1	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.

Navigation buttons at the bottom: [Cancel](#), [Previous](#), [Review and Launch](#), [Next: Add Tags](#).

4. Configuring Security Group

The screenshot shows the AWS Management Console at the same URL as the previous step. The navigation bar and progress bar are consistent, with '6. Configure Security Group' selected.

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

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

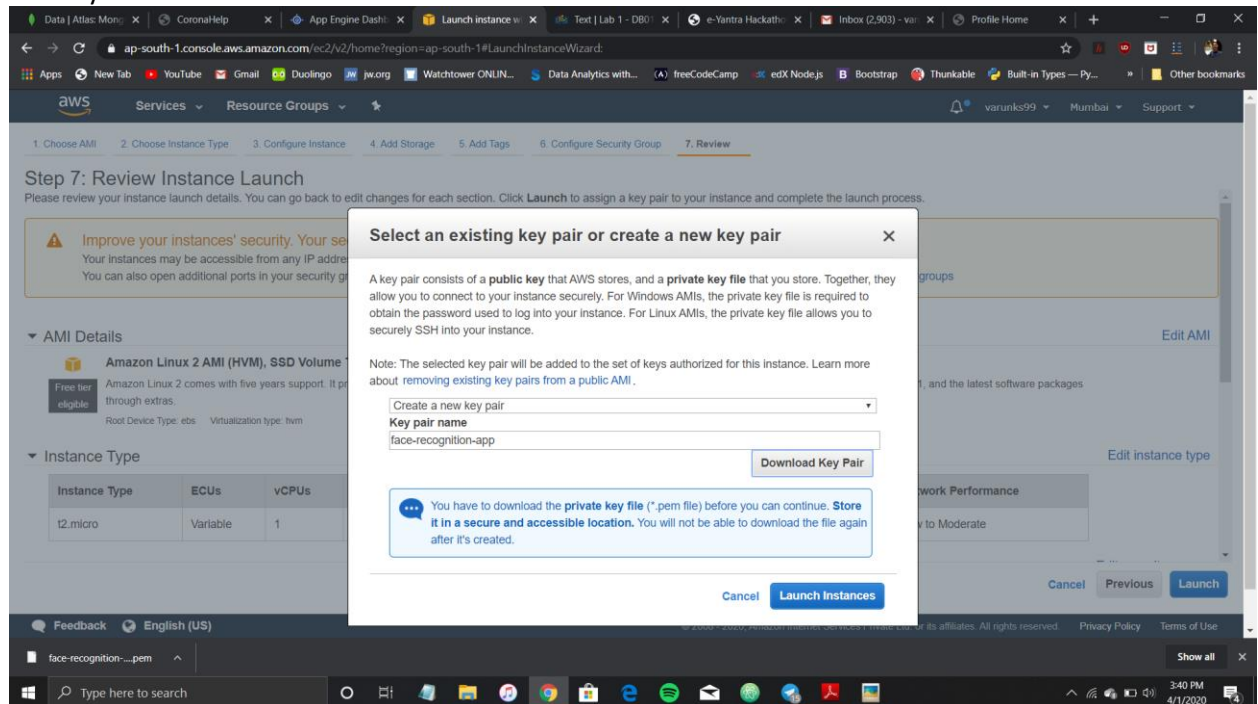
[Add Rule](#)

Warning

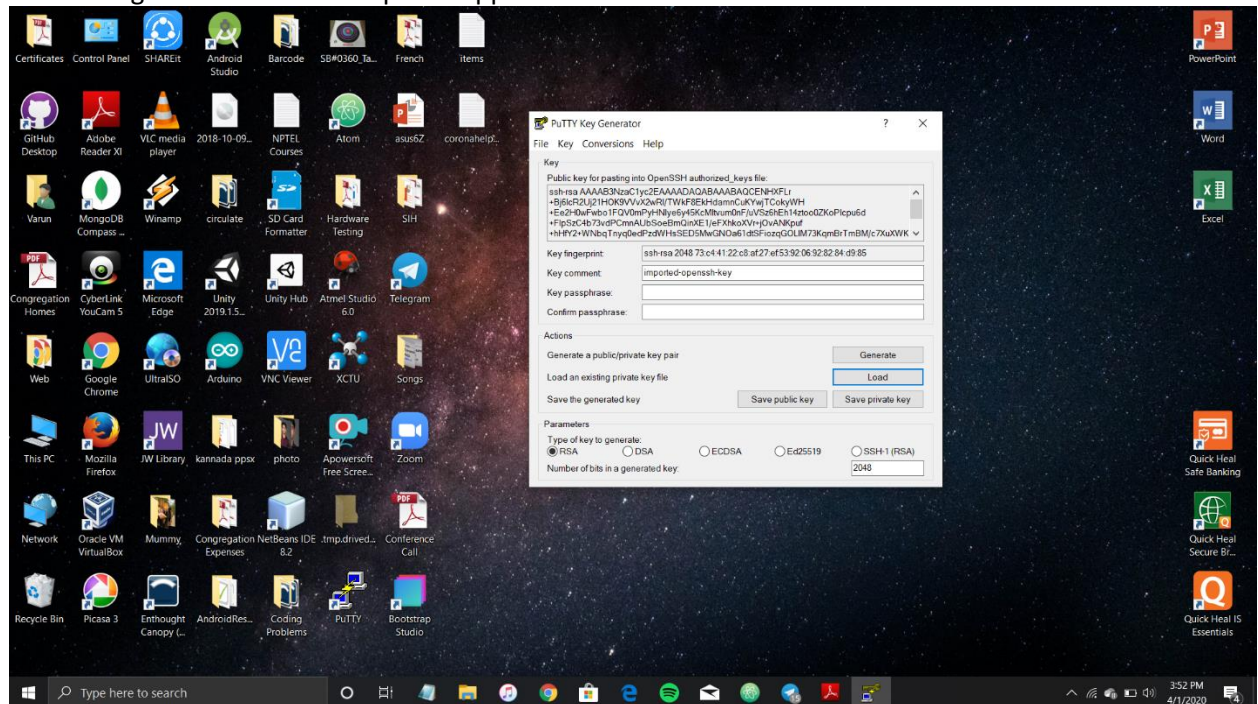
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Navigation buttons at the bottom: [Cancel](#), [Previous](#), [Review and Launch](#).

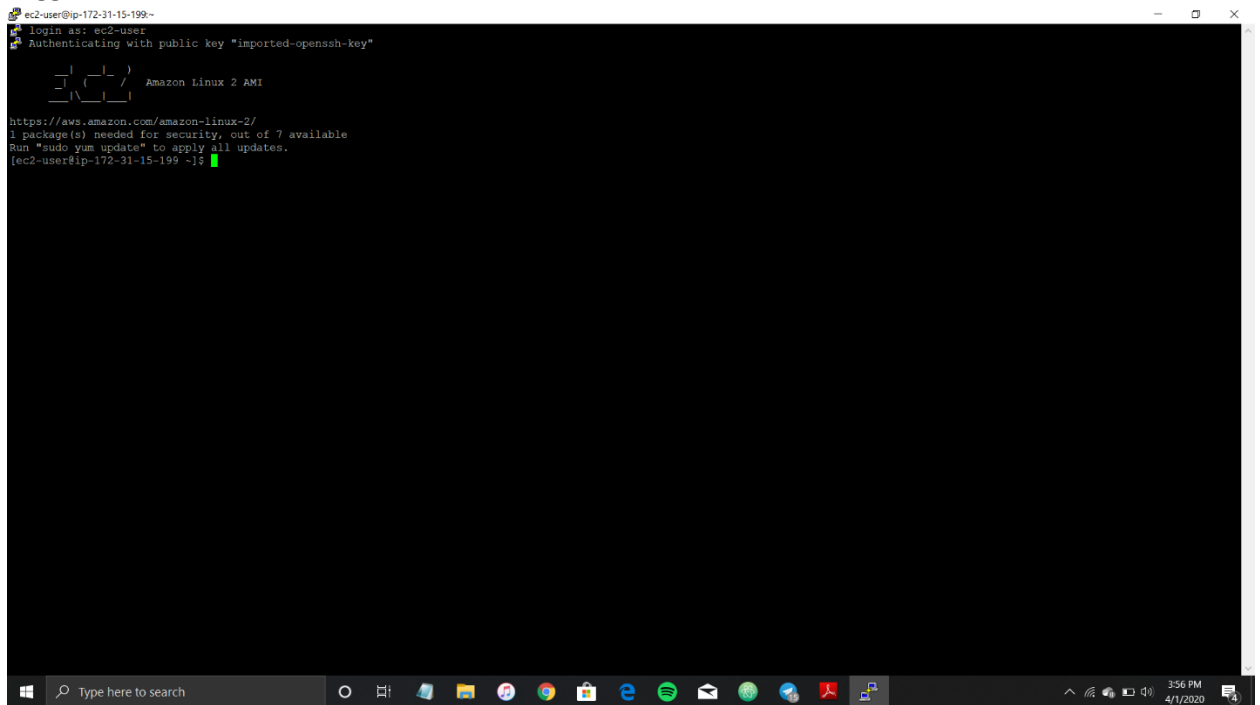
5. Key Pair Download



6. PuTTYgen conversion from pem to ppk

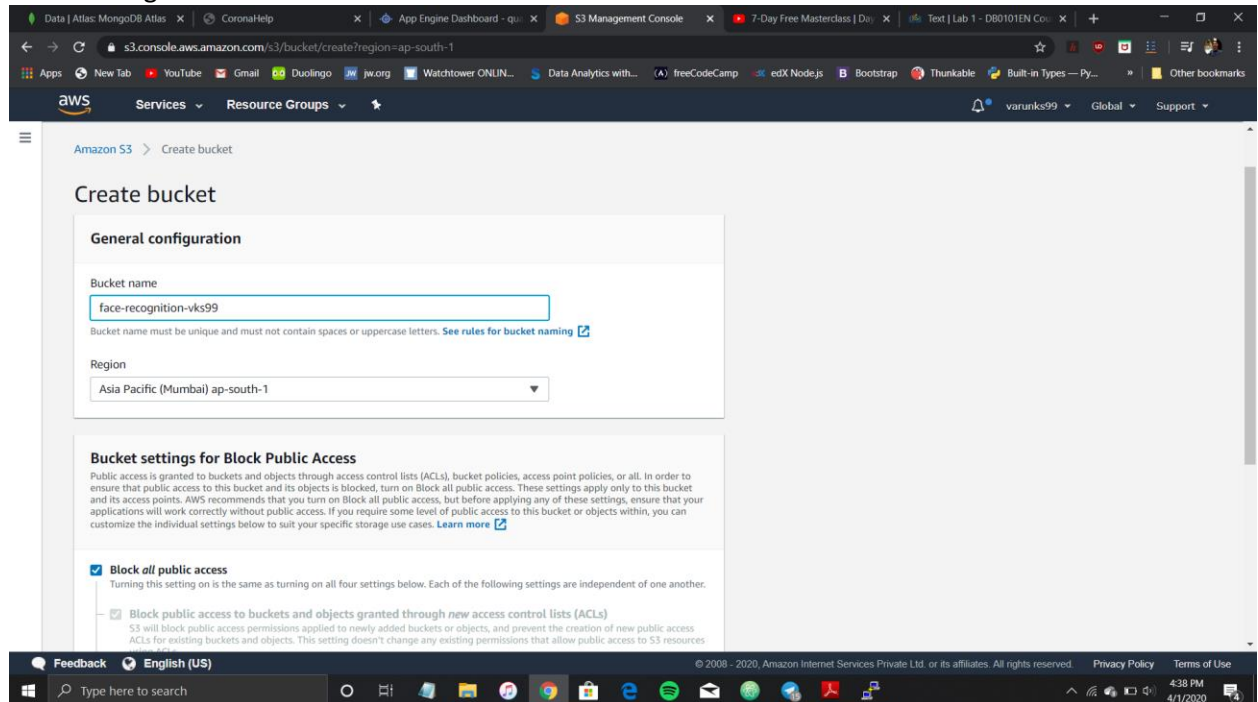


7. Logged in EC2 black screen

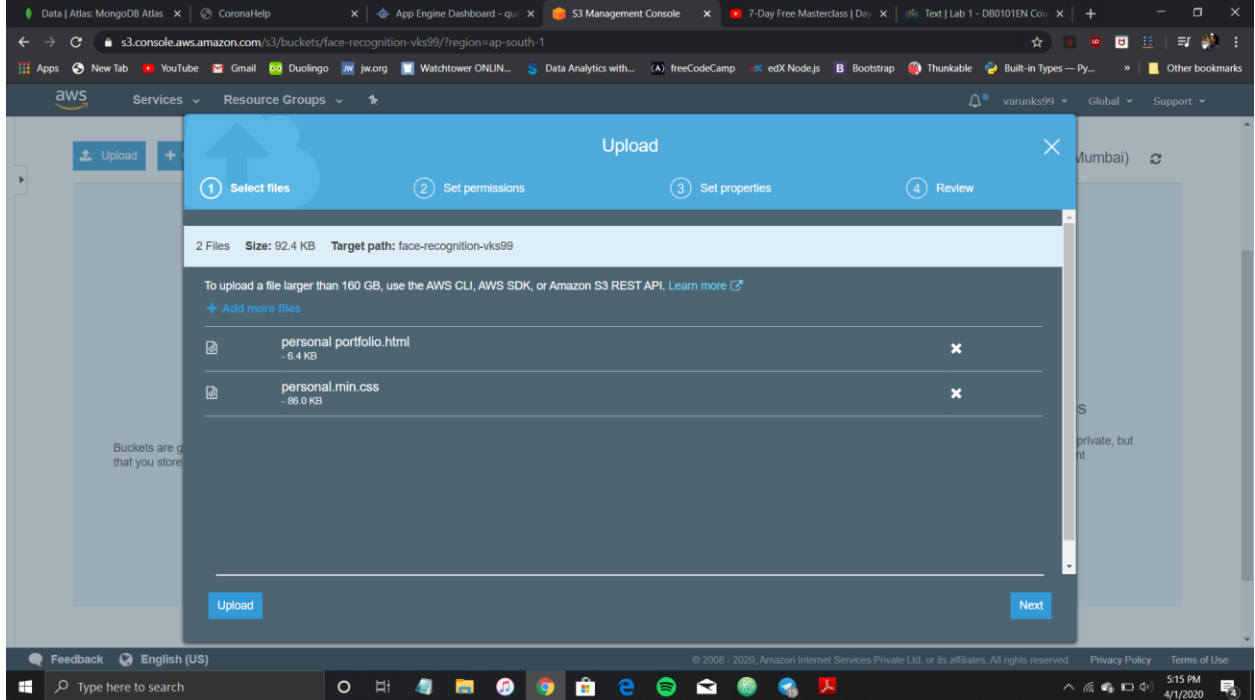


S3

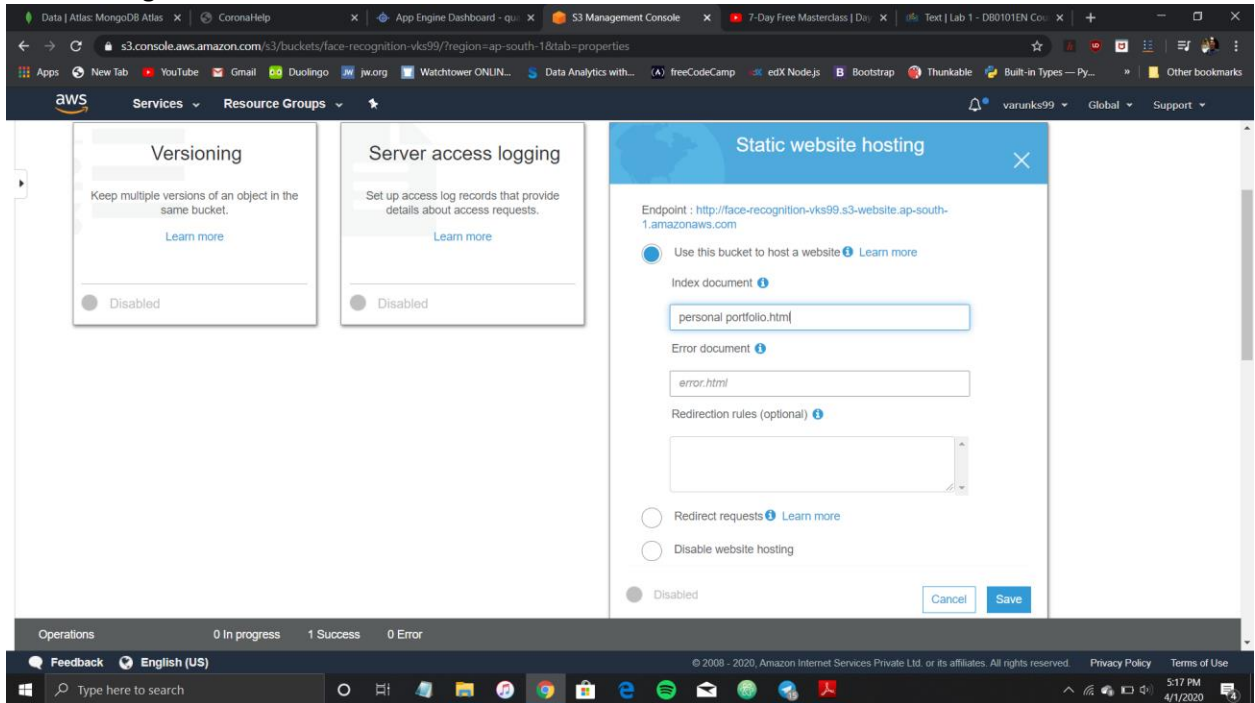
1. Creating a bucket



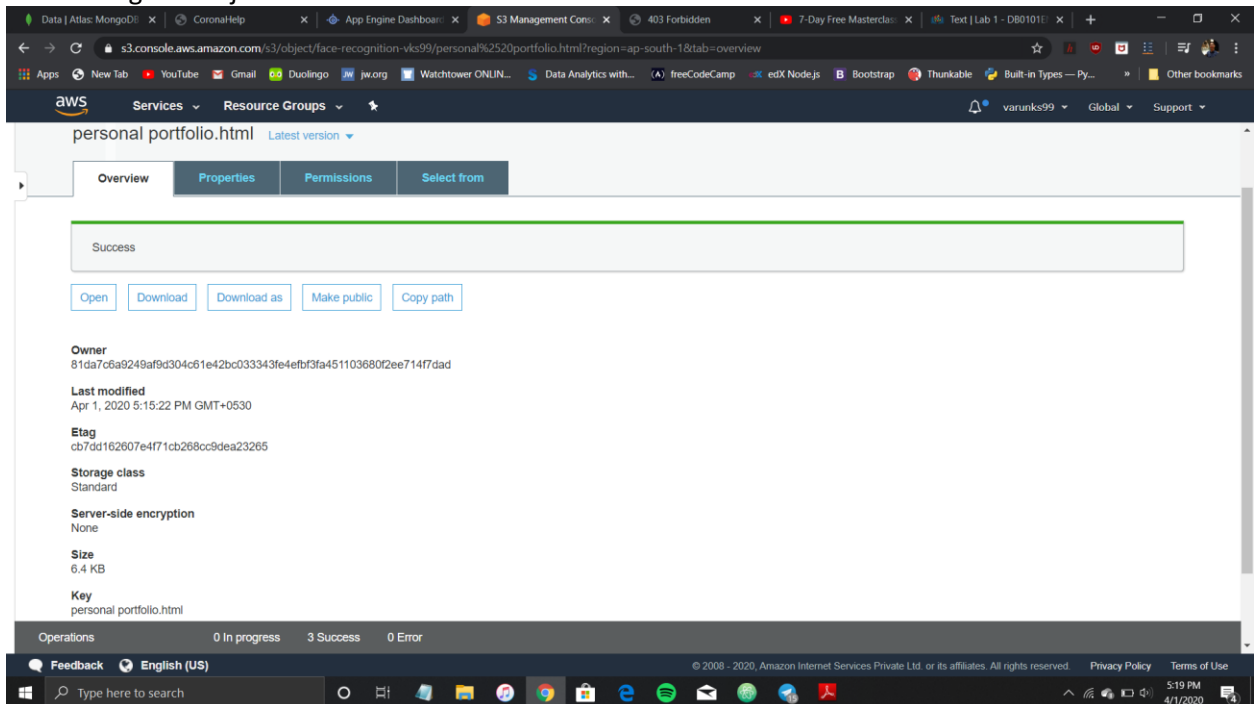
2. Uploading an Object



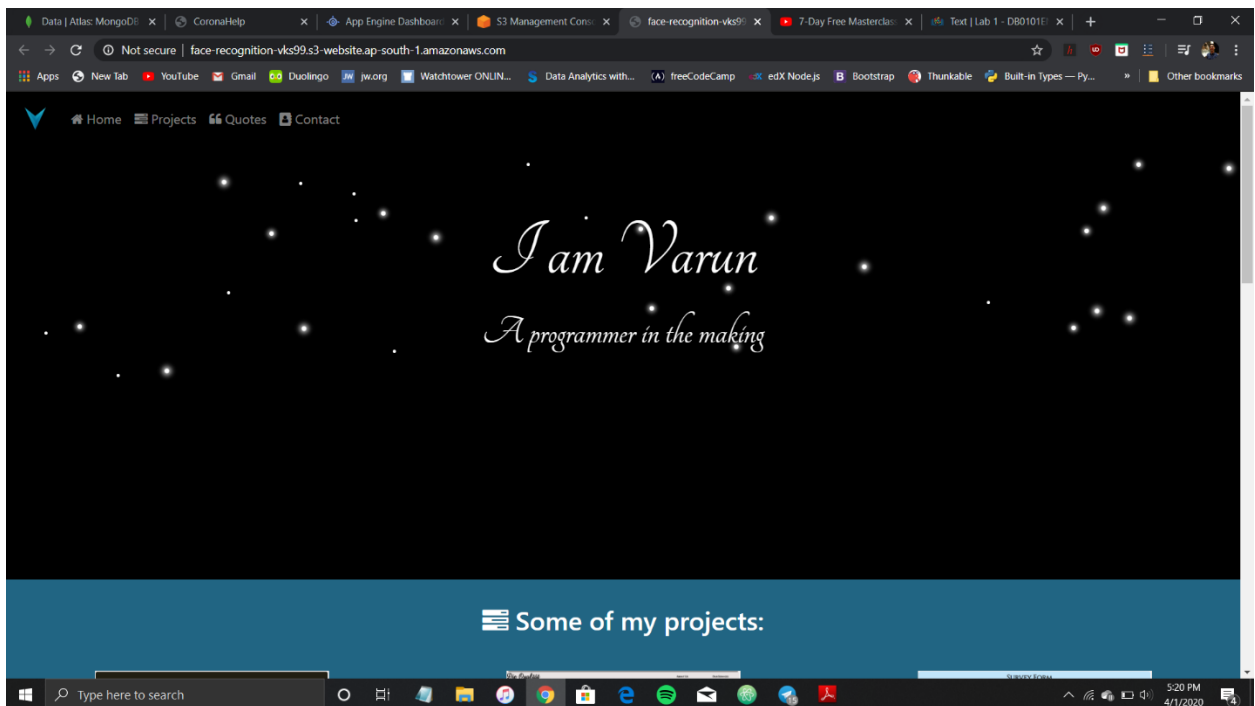
3. Enabling Static Website



4. Making the Object Public

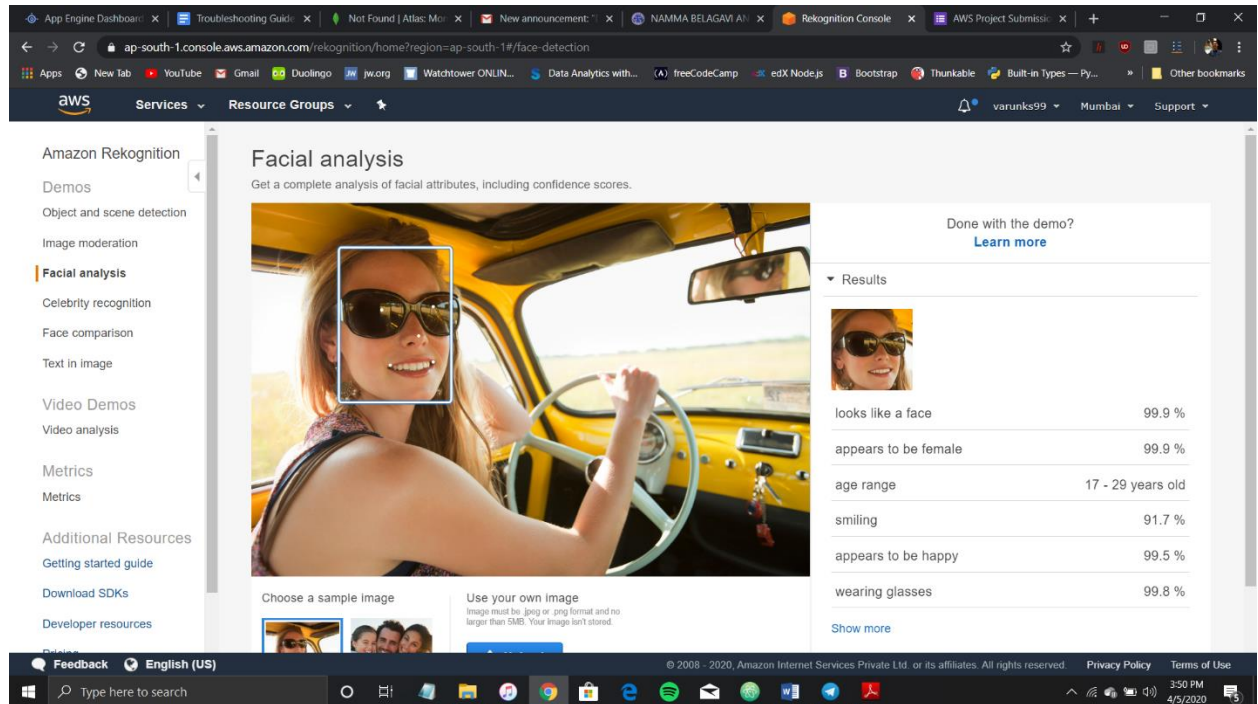


5. Checking the S3 link on the browser



Rekognition

1. Face Detect



Amazon Rekognition

Demos

- Object and scene detection
- Image moderation
- Facial analysis**
- Celebrity recognition
- Face comparison
- Text in image

Video Demos

- Video analysis


Metrics

Additional Resources

- Getting started guide
- Download SDKs
- Developer resources

Facial analysis

Get a complete analysis of facial attributes, including confidence scores.



Choose a sample image

Use your own image
Image must be .jpg or .png format and no larger than 5MB. Your image isn't stored.

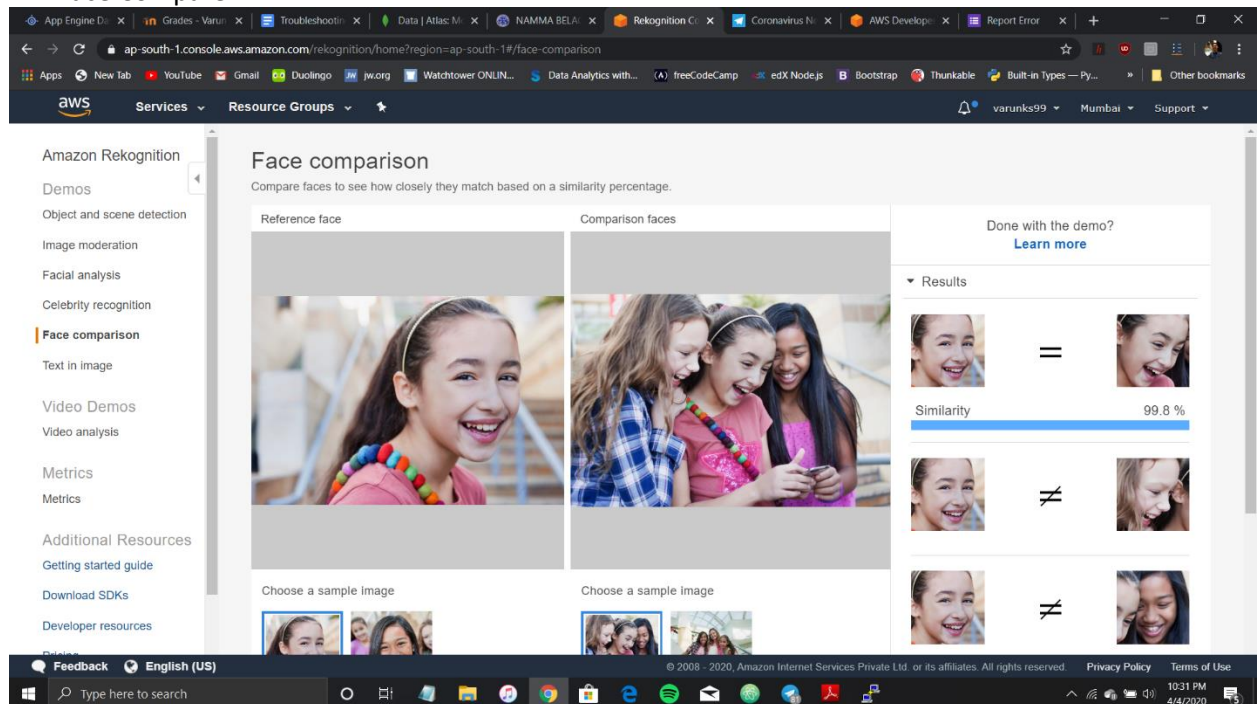
Done with the demo? [Learn more](#)

Results

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 %

Show more

2. Face Compare



Amazon Rekognition

Demos

- Object and scene detection
- Image moderation
- Facial analysis
- Celebrity recognition
- Face comparison**
- Text in image

Video Demos

- Video analysis

Metrics


Additional Resources

- Getting started guide
- Download SDKs
- Developer resources


Face comparison

Compare faces to see how closely they match based on a similarity percentage.

Reference face



Comparison faces









Choose a sample image

Choose a sample image

Done with the demo? [Learn more](#)

Results

	=	
Similarity 99.8 %		
	≠	
	≠	

3. Celebrity Recognition

The screenshot shows the AWS Rekognition console in the 'Celebrity recognition' demo. The left sidebar lists various services, with 'Celebrity recognition' selected. The main area displays a photo of Jeff Bezos with a bounding box around his face. Below the photo are options to 'Choose a sample image' or 'Use your own image'. On the right, the 'Results' section shows a match for 'Jeff Bezos' with a 'Match confidence' of 100%. The bottom of the browser window shows the Windows taskbar with the time 10:31 PM on 4/4/2020.

Amazon Rekognition

Demos

- Object and scene detection
- Image moderation
- Facial analysis
- Celebrity recognition**
- Face comparison
- Text in image

Video Demos

- Video analysis

Metrics

- Metrics

Additional Resources


- Getting started guide
- Download SDKs
- Developer resources

Celebrity recognition

Rekognition automatically recognizes celebrities in images and provides confidence scores.

Done with the demo? [Learn more](#)

▼ Results

 **Jeff Bezos**
[Learn More](#)

Match confidence 100 %

► Request

► Response

Choose a sample image

Use your own image
Image must be .jpeg or .png format and no larger than 5MB. Your image isn't stored.

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

10:31 PM 4/4/2020

4. Text in Image

The screenshot shows the AWS Rekognition console in the 'Text in image' demo. The left sidebar lists various services, with 'Text in image' selected. The main area displays an image of a coffee cup with the text 'IT'S MONDAY but keep Smiling' overlaid. Below the image are options to 'Choose a sample image' or 'Use your own image'. On the right, the 'Results' section shows the detected text: 'IT'S MONDAY but keep Smiling'. The bottom of the browser window shows the Windows taskbar with the time 10:31 PM on 4/4/2020.

Amazon Rekognition

Demos

- Object and scene detection
- Image moderation
- Facial analysis
- Celebrity recognition
- Face comparison
- Text in image**

Video Demos

- Video analysis

Metrics

- Metrics

Additional Resources

- Getting started guide
- Download SDKs
- Developer resources

Text in image

Rekognition automatically detects and extracts text in your images. [Learn More](#)

Done with the demo? [Learn more](#)

▼ Results

US English only

| IT'S |
| MONDAY |
| but | keep |
| Smiling |

► Request

► Response

Choose a sample image

Use your own image
Image must be .jpeg or .png format and no larger than 5MB. Your image isn't stored.

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

10:31 PM 4/4/2020

1. Installing aws-sdk

```

ec2-user@ip-172-31-9-199:var/www/html%
- Installing symfony/event-dispatcher (v2.8.52): Downloading (100%)
  proc_open(): fork failed - Cannot allocate memory
  The archive may contain identical file names with different capitalization (which fails on case insensitive filesystems)
  Unzip with unzip command failed, falling back to ZipArchive class

Installation failed, deleting ./composer.json
The following exception is caused by a lack of memory or swap, or not having swap configured
Check https://getcomposer.org/doc/articles/troubleshooting.md#proc-open-fork-failed-errors for details

PHP Warning:  proc_open(): fork failed - Cannot allocate memory in phar:///home/ec2-user/composer.phar/vendor/symfony/console/Application.php on line 952
Warning: proc_open(): fork failed - Cannot allocate memory in phar:///home/ec2-user/composer.phar/vendor/symfony/console/Application.php on line 952

[ErrorException]
proc_open(): fork failed - Cannot allocate memory

[ec2-user@ip-172-31-9-199 face]$ 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, 13.3485 s, 80.4 MB/s
[ec2-user@ip-172-31-9-199 face]$ sudo /sbin/mkswap /var/swap.1
mkswap: /var/swap.1: insecure permissions 0644, 0600 suggested.
Setting up swapspace version 1, size = 1024 MB (1073737728 bytes)
no label, UUID=f857b711-b3d4-4cc8-b14e-77f1b12b06da
[ec2-user@ip-172-31-9-199 face]$ sudo /sbin/swapon /var/swap.1
swapon: /var/swap.1: insecure permissions 0644, 0600 suggested.
[ec2-user@ip-172-31-9-199 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 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 (v3.9.2): 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-172-31-9-199 face]$

```

```
ec2-user@ip-172-31-9-199:~$
login as: ec2-user
Authenticating with public key "imported-openssh-key"

    ____  _
   / ___/  | |__  _ __ ___  | |
  / /   / __/ _  / _  /   | |
 / ___/ /_/ / ___/ /___/   | |
/_/___/____/_/_____/_____|_|

Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
1 package(s) needed for security, out of 7 available
Run "sudo yum update" to apply all updates.
ec2-user@ip-172-31-9-199 ~$ sudo yum install php
loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                               | 2.4 kB  00:00:00
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: php-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
--> Processing Dependency: httpd-mmn = 20120211x8664 for package: php-5.4.16-46.amzn2.0.2.x86_64
--> Processing Dependency: httpd for package: php-5.4.16-46.amzn2.0.2.x86_64
--> Running transaction check
--> Package httpd.x86_64 0:2.4.41-1.amzn2.0.1 will be installed
--> Processing Dependency: httpd-tools = 2.4.41-1.amzn2.0.1 for package: httpd-2.4.41-1.amzn2.0.1.x86_64
--> Processing Dependency: httpd-filesystem = 2.4.41-1.amzn2.0.1 for package: httpd-2.4.41-1.amzn2.0.1.x86_64
--> Processing Dependency: system-logos-httpd for package: httpd-2.4.41-1.amzn2.0.1.x86_64
--> Processing Dependency: mod_http2 for package: httpd-2.4.41-1.amzn2.0.1.x86_64
--> Processing Dependency: httpd-filesystem for package: httpd-2.4.41-1.amzn2.0.1.x86_64
--> Processing Dependency: /etc/mime.types for package: httpd-2.4.41-1.amzn2.0.1.x86_64
--> Processing Dependency: libaprutil1.so.0()(64bit) for package: httpd-2.4.41-1.amzn2.0.1.x86_64
--> Processing Dependency: libapr-1.so.0()(64bit) for package: httpd-2.4.41-1.amzn2.0.1.x86_64
--> 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-5.4.16-46.amzn2.0.2.x86_64
--> Running transaction check
--> Package apr.x86_64 0:1.6.3-5.amzn2.0.2 will be installed
--> Package apr-util.x86_64 0:1.6.1-5.amzn2.0.2 will be installed
--> Processing Dependency: apr-util-bdb(x86-64) = 1.6.1-5.amzn2.0.2 for package: apr-util-1.6.1-5.amzn2.0.2.x86_64
--> Package generic-logos-httpd.noarch 0:18.0-0.4.amzn2 will be installed
--> Package httpd-filesystem.noarch 0:2.4.41-1.amzn2.0.1 will be installed
--> Package httpd-tools.x86_64 0:2.4.41-1.amzn2.0.1 will be installed
--> Package libzip010-compat.x86_64 0:0.10.1-9.amzn2.0.5 will be installed
--> Package mailcap.noarch 0:2.1.41-2.amzn2 will be installed
--> Package mod_http2.x86_64 0:1.15.3-2.amzn2 will be installed
--> Running transaction check
--> Package apr-util-bdb.x86_64 0:1.6.1-5.amzn2.0.2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved
```

3. index.php file code

```
ec2-user@ip-172-31-9-199:/var/www/html/face
use Aws\S3\S3Client;
use Aws\Rekognition\RekognitionClient;

$bucket = 'face-recognition-vks99';
$keyname = 'sample.jpg';

$s3 = new S3Client([
    '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' => 'us-east-2',
            '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;
}

-- INSERT --
```

4. Upload success screenshot

```
ec2-user@ip-172-31-8-255:/var/www/html/face
[ec2-user@ip-172-31-8-255 face]$ sudo php index.php
Image upload done... Here is the URL: https://face-recognition-vks99.s3.ap-south-1.amazonaws.com/sample.jpgPHP Fatal error:  Uncaught Error: Class 'Aws\Rekognition\RekognitionClient' not found in /var/www/html/face/index.php:63
Stack trace:
#0 (main):
thrown in /var/www/html/face/index.php on line 63
[ec2-user@ip-172-31-8-255 face]$
```