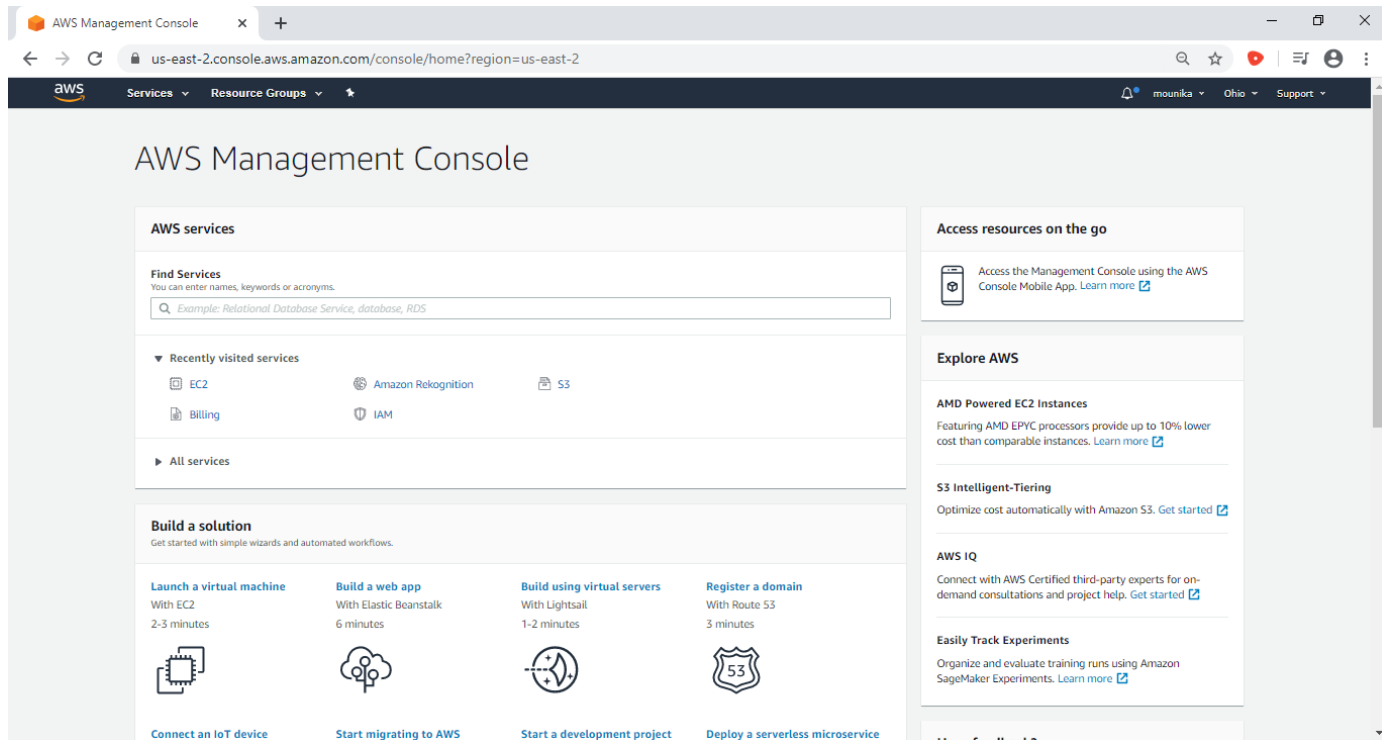


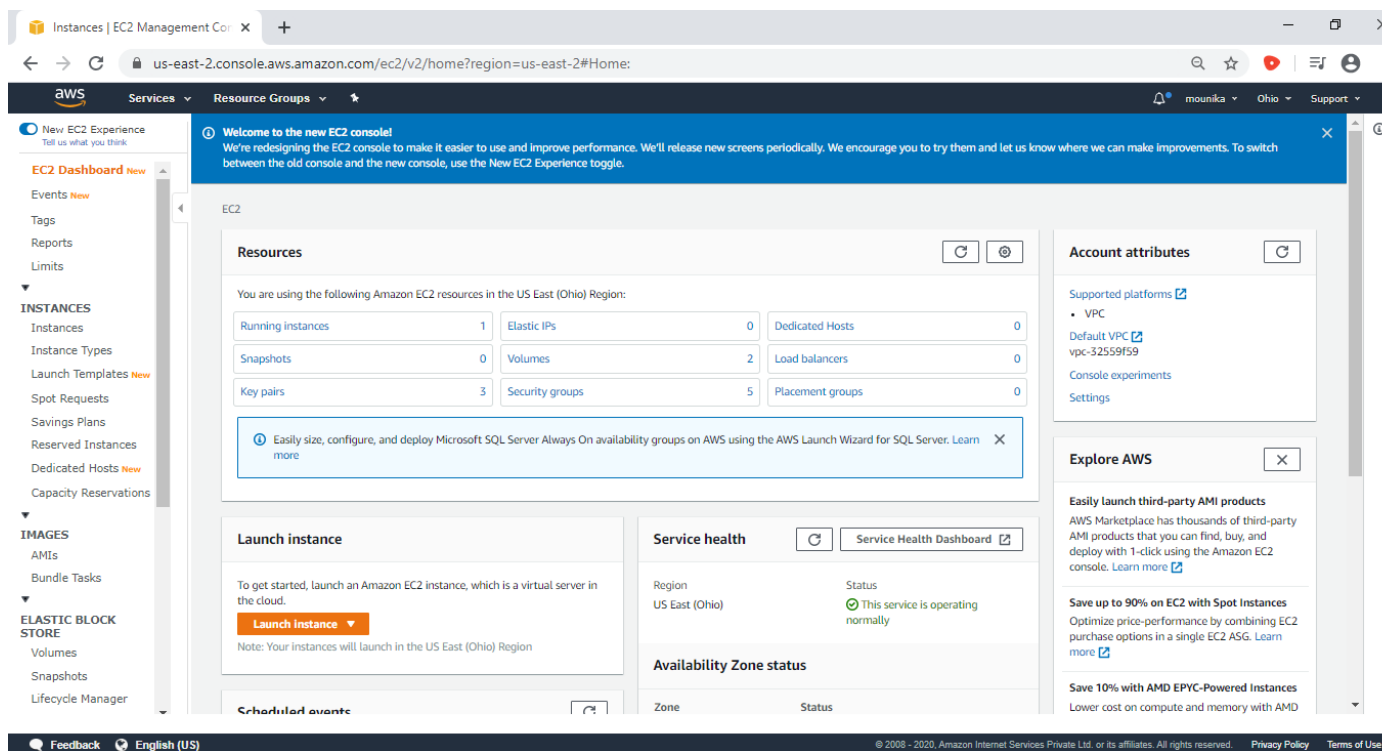
Number of Faces Detection using AWS Rekognition

-By Y. Mounika

1. AWS Login screen with username



2. EC2 Dashboard



3. S3 Dashboard

The screenshot shows the Amazon S3 Management Console. The left sidebar contains navigation links: Buckets, Batch operations, Access analyzer for S3, Block public access (account settings), and Feature spotlight (2). The main content area is titled 'Amazon S3' and shows 'Buckets (1)'. A search bar 'Find bucket by name' is present. Below it is a table with columns: Name, Region, Access, and Bucket created. One bucket is listed: 'myproject1-mounika' in the 'us-east-2' region, created on '2020-03-27T06:10:04.000Z'. Action buttons 'Copy ARN', 'Empty', 'Delete', and 'Create bucket' are at the top right. The footer includes 'Feedback', 'English (US)', and copyright information.

Name	Region	Access	Bucket created
myproject1-mounika	us-east-2		2020-03-27T06:10:04.000Z

4. Rekognition Dashboard

The screenshot shows the Amazon Rekognition Console. The left sidebar lists navigation options: Custom Labels (New), Use Custom Labels, 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, Pricing, FAQ). The main content area features a large hero section with the title 'Amazon Rekognition' and subtitle 'Deep learning-based visual analysis service'. It includes a 'Try Demo' button and a 'Download SDKs' link. Below the hero section are three columns: 'Easily Integrate Powerful Visual Analysis into Your App', 'Continuously Learning', and 'Integrated with AWS Services'. The footer includes 'Feedback', 'English (US)', and copyright information.

Amazon Rekognition
Deep learning-based visual analysis service
Search, verify, and organize millions of images and videos
[Try Demo](#)
[Download SDKs](#)

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 visual recognition capabilities by making powerful and accurate analysis available with easy to use APIs.

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 without moving any data. You can also run real-time video analysis on streams coming from Amazon Kinesis Video Streams.

Screenshots for EC2

1. Choosing an AMI

Launch instance wizard | EC2 Ma x +

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

aws 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

Cancel and Exit

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 Free tier eligible

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

64-bit (x86)

64-bit (Arm)

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

Amazon Linux Free tier eligible

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.

64-bit (x86)

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

2. Choosing an Instance Type

Launch instance wizard | EC2 Ma x +

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

aws 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

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 Free tier eligible	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

Cancel Previous Review and Launch Next: Configure Instance Details

3. Adding Storage

Launch instance wizard | EC2 Ma

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

aws 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 Encrypt

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

4. Configuring Security Group

Launch instance wizard | EC2 Ma

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

aws 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 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: launch-wizard-2

Description: launch-wizard-2 created 2020-03-26T23:23:50.850+05:30

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.

Cancel Previous Review and Launch

5. Key Pair Download

Launch instance wizard | EC2 Ma x +

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

aws 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 7: Review Instance Launch

AMI Details

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0e01ce4ee18447327

Free tier eligible

Root Device Type: ebs Virtualization type: hvm

Instance Type

Instance Type	ECUs	vCPUs	Memory (GiB)
t2.micro	Variable	1	1

Security Groups

Security group name: launch-wizard-2
Description: launch-wizard-2 created 2020-03-26T23:23:50.85

Type	Protocol
SSH	TCP

Instance Details

Storage

Tags

Select an existing key pair or create a new key pair

A key pair consists of a public key that AWS stores, and a private key file that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.

Create a new key pair

Key pair name: project1-key

Download Key Pair

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.

Cancel Launch Instances

Feedback English (US)

© 2008 - 2020 Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

project1-key2.pem

Show all

6. PuTTYgen conversion from pem to ppk

PuTTY Key Generator

File Key Conversions Help

Key

Public key for pasting into OpenSSH authorized_keys file:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQCE3qpvPqjO/V6Lsnr0OwVXQsKz0l7bQrzt
5ZZ1TY3Xmv9cJU/giGEELf47Mt0CQsMjFSNNL5BcqtzpAelYozGANKJKCuNlw4/2rEC
20v83lrOS7WBz0YYgov6OjpGIQIf
+z/tvZxxd4pdeXDLT0BfmZL5npSA4VyAdeSrw6Uzqk7dWHPz8xhonYwLR
```

Key fingerprint: ssh-rsa 2048 e1:a0:48:c4:08:d7:8b:25:a6:59:0f:f5:c7:0d:8e:11

Key comment: imported-openssh-key

Key passphrase:

Confirm passphrase:

Actions

Generate a public/private key pair Generate

Load an existing private key file Load

Save the generated key Save public key Save private key

Parameters

Type of key to generate:
☒ RSA ☐ DSA ☐ ECDSA ☐ Ed25519 ☐ SSH-1 (RSA)

Number of bits in a generated key: 2048

7. Logged in EC2 black screen

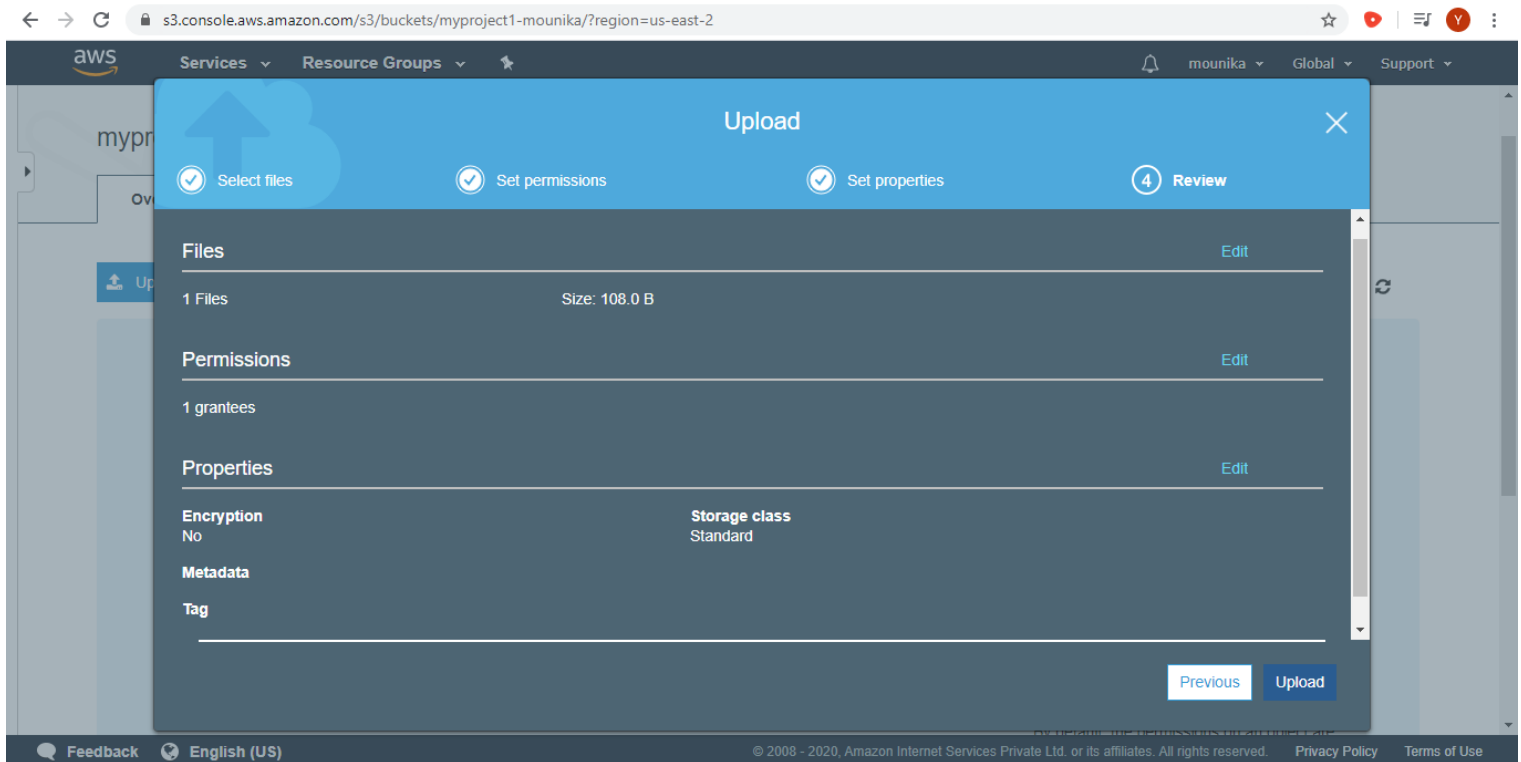
The screenshot shows the AWS Management Console for the 'us-east-2' region. A terminal window is open for an EC2 instance named 'i-0cb31cbb78bf1cbb7'. The terminal output shows a successful login as 'ec2-user' using a public key. The prompt is 'ec2-user@ip-172-31-35-66:~'. The terminal also displays the Amazon Linux 2 AMI logo and a message about security updates: '1 package(s) needed for security, out of 7 available. Run "sudo yum update" to apply all updates.' The instance details on the left show it is a 't2.micro' instance with a 'project1-key' key pair. The instance state is 'running'.

Screenshots needed for S3

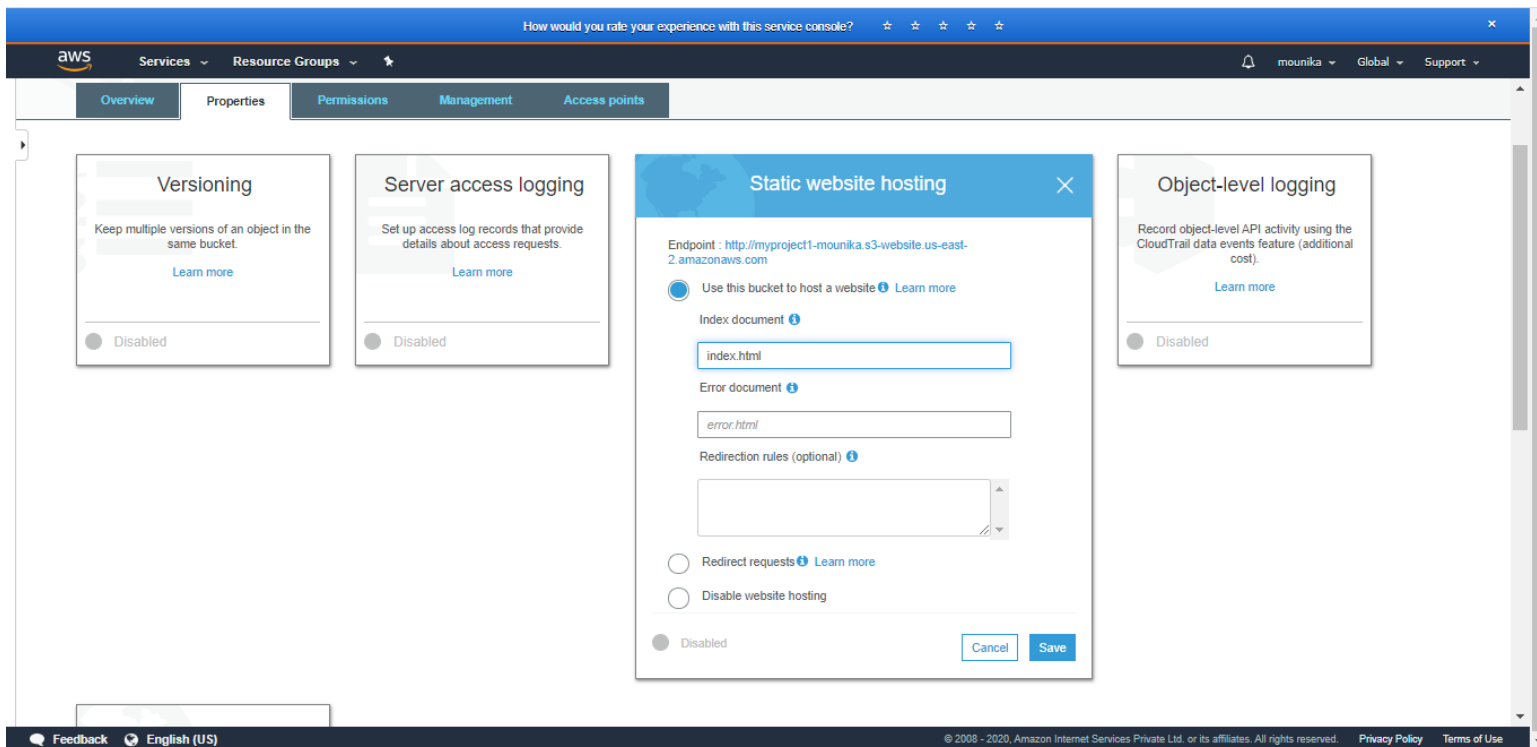
1. Creating a bucket

The screenshot shows the 'Create bucket' page in the AWS S3 console. The 'General configuration' section shows the bucket name 'myproject1-mounika' and the region 'US East (Ohio) us-east-2'. The 'Bucket settings for Block Public Access' section is expanded, showing that 'Block all public access' is checked. The settings include: '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 and cross-account access to buckets and objects through any public bucket or access point'.

2. Uploading an Object



3. Enabling Static Website

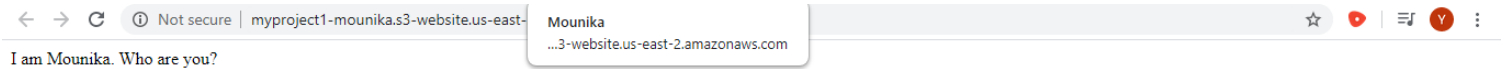


The screenshot shows the AWS IAM console interface for a bucket named 'myproject1-mounika'. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', and user information 'mounika'. The main content area has tabs for 'Overview', 'Properties', 'Permissions', 'Management', and 'Access points'. Under the 'Management' tab, there are five settings cards: 'Versioning' (Disabled), 'Server access logging' (Disabled), 'Static website hosting' (Bucket hosting is selected), 'Object-level logging' (Disabled), and 'Default encryption' (Disabled). Below these is the 'Advanced settings' section with five more cards: 'Object lock' (Disabled), 'Tags' (0 Tags), 'Transfer acceleration' (Suspended), 'Events' (0 Active notifications), and 'Requester pays' (Disabled). Each card includes a brief description and a 'Learn more' link. The footer contains 'Feedback', 'English (US)', and copyright information.

4. Making the Object Public

The screenshot shows the AWS S3 console interface for the 'index.html' object in the 'myproject1-mounika' bucket. The breadcrumb trail is 'Amazon S3 > myproject1-mounika > index.html'. The 'index.html' tab is selected, showing a 'Success' message at the top. Below the message are buttons for 'Open', 'Download', 'Download as', 'Make public', and 'Copy path'. The object details section lists the following information: Owner (8956fcd98911da4dc3cd015476af9b17055f3eba8c4f106d17eb1575fce9c12), Last modified (Mar 27, 2020 11:45:20 AM GMT+0530), Etag (423c3e240774a3af2d785273e6bcb4c8), Storage class (Standard), Server-side encryption (None), Size (108.0 B), Key (index.html), and Object URL (https://myproject1-mounika.s3.us-east-2.amazonaws.com/index.html). The bottom status bar shows '0 In progress', '1 Success', and '0 Error' operations. The footer includes 'Feedback', 'English (US)', and copyright information.

5. Checking the S3 link on the browser



Screenshots for Rekognition

1. Face Detect

The screenshot shows the AWS Rekognition 'Facial analysis' demo page. The main area displays a photo of a woman with a bounding box around her face. To the right, a 'Results' section lists attributes and their confidence scores:

Attribute	Confidence Score
looks like a face	99.9 %
appears to be female	54.8 %
age range	22 - 34 years old
smiling	54.9 %
appears to be happy	54.9 %
not wearing glasses	54.9 %

Below the main image, there are two options for image input: 'Choose a sample image' (with a button to view samples) and 'Use your own image' (with an 'Upload' button and a text field for 'Use image URL'). A 'Go' button is at the bottom right of the input section.

2. Face Compare

us-east-2.console.aws.amazon.com/rekognition/home?region=us-east-2#/face-comparison

Amazon Rekognition

Custom Labels ^{New}

Use Custom Labels

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


Pricing

FAQ


Face comparison

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


Reference face



Comparison faces



Choose a sample image



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

▼ Results

Similarity 99 %

Request

Response

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

3. Celebrity Recognition

us-east-2.console.aws.amazon.com/rekognition/home?region=us-east-2#/celebrity-detection

Amazon Rekognition

Services

Resource Groups


Feedback

English (US)

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

Celebrity recognition

Rekognition automatically recognizes celebrities in images and provides confidence scores.



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

▼ Results

Virat Kohli

Match confidence 100 %

Request

Response

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

4. Text in Image


us-east-2.console.aws.amazon.com/rekognition/home?region=us-east-2#/text-detection

aws Services Resource Groups



mounika Ohio Support

Text in image

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



Choose a sample image



Use your own image

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

Upload or drag and drop

Go

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

ResultsUS English only

| SURF |

Request

Response

Screenshots for EC2 & S3

1. Installing aws-sdk

```
ec2-user@ip-172-31-42-123:/var/www/html/face
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

require [--dev] [--prefer-source] [--prefer-dist] [--fixed] [--no-progress] [--no-suggest] [--no-update] [--no-scripts] [--update-no-dev] [--update-with-dependencies] [
--update-with-all-dependencies] [--ignore-platform-reqs] [--prefer-stable] [--prefer-lowest] [--sort-packages] [-o|--optimize-autoloader] [-a|--classmap-authoritative]
[--apcu-autoloader] [--] [<packages>]...

[ec2-user@ip-172-31-42-123 face]$ sudo /bin/dd if=/dev/zero of=/var/swap.1 bs=1M co
unt=1024
1024+0 records in
1024+0 records out
1073741824 bytes (1.1 GB) copied, 13.3657 s, 80.3 MB/s
[ec2-user@ip-172-31-42-123 face]$ sudo /sbin/mkswap /var/swap.1
mkswap: /var/swap.1: insecure permissions 0644, 0600 suggested.
Setting up swapspace version 1, size = 1024 MiB (1073737728 bytes)
no label, UUID=288f3cca-bdbc-4179-a590-9cd61fee9f1a
[ec2-user@ip-172-31-42-123 face]$ sudo /sbin/swapon /var/swap.1
swapon: /var/swap.1: insecure permissions 0644, 0600 suggested.
[ec2-user@ip-172-31-42-123 face]$ sudo php -d memory_limit=-1 ~/composer.phar requi
re 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
[ec2-user@ip-172-31-42-123 face]$
```

If the command isn't clear in the above screenshot, I have cropped it for better visibility in the below screen shot.

```
[ec2-user@ip-172-31-42-123 face]$ sudo php -d memory_limit=-1 ~/composer.phar requi
re 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
[ec2-user@ip-172-31-42-123 face]$
```

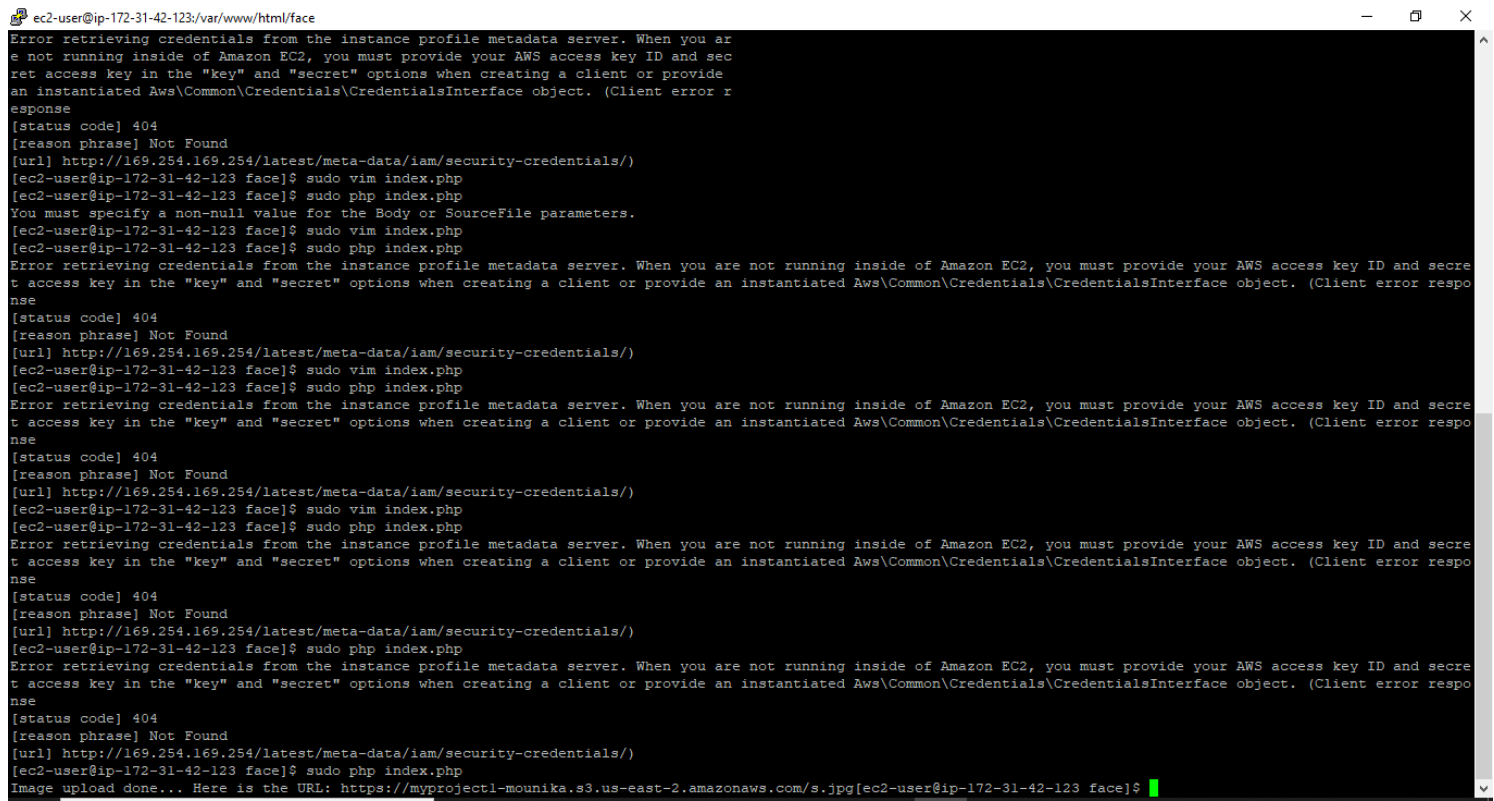
2. Installing php

```
ec2-user@ip-172-31-42-123:~  
[ec2-user@ip-172-31-42-123 ~]$ 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-172-31-42-123 ~]$
```

2. index.php file code

```
ec2-user@ip-172-31-42-123:/var/www/html/face  
<?php  
require_once(__DIR__ . '/vendor/autoload.php');  
  
use Aws\S3\S3Client;  
use Aws\Rekognition\RekognitionClient;  
$bucket = 'myproject1-mounika';  
$keyname = 's.jpg';  
$s3 = S3Client::factory([  
    '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',  
        ]);  
        echo "RekognitionClient is working";  
  
        $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;  
}
```

3. Upload success screenshot

A terminal window titled 'ec2-user@ip-172-31-42-123:/var/www/html/face' showing a series of commands and error messages. The user attempts to run 'sudo php index.php' multiple times, but receives a 404 error from the AWS CLI. The error message states: 'Error retrieving credentials from the instance profile metadata server. When you are not running inside of Amazon EC2, you must provide your AWS access key ID and secret access key in the "key" and "secret" options when creating a client or provide an instantiated Aws\Common\Credentials\CredentialsInterface object. (Client error response [status code] 404 [reason phrase] Not Found [url] http://169.254.169.254/latest/meta-data/iam/security-credentials/)' The user then runs 'sudo vim index.php' and 'sudo php index.php' again, but the same error occurs. Finally, the user runs 'Image upload done... Here is the URL: https://myproject1-mounika.s3.us-east-2.amazonaws.com/s.jpg' and the terminal shows a green cursor at the end of the line.

```
ec2-user@ip-172-31-42-123:/var/www/html/face
Error retrieving credentials from the instance profile metadata server. When you are not running inside of Amazon EC2, you must provide your AWS access key ID and secret access key in the "key" and "secret" options when creating a client or provide an instantiated Aws\Common\Credentials\CredentialsInterface object. (Client error response [status code] 404 [reason phrase] Not Found [url] http://169.254.169.254/latest/meta-data/iam/security-credentials/)
[ec2-user@ip-172-31-42-123 face]$ sudo vim index.php
[ec2-user@ip-172-31-42-123 face]$ sudo php index.php
You must specify a non-null value for the Body or SourceFile parameters.
[ec2-user@ip-172-31-42-123 face]$ sudo vim index.php
[ec2-user@ip-172-31-42-123 face]$ sudo php index.php
Error retrieving credentials from the instance profile metadata server. When you are not running inside of Amazon EC2, you must provide your AWS access key ID and secret access key in the "key" and "secret" options when creating a client or provide an instantiated Aws\Common\Credentials\CredentialsInterface object. (Client error response [status code] 404 [reason phrase] Not Found [url] http://169.254.169.254/latest/meta-data/iam/security-credentials/)
[ec2-user@ip-172-31-42-123 face]$ sudo vim index.php
[ec2-user@ip-172-31-42-123 face]$ sudo php index.php
Error retrieving credentials from the instance profile metadata server. When you are not running inside of Amazon EC2, you must provide your AWS access key ID and secret access key in the "key" and "secret" options when creating a client or provide an instantiated Aws\Common\Credentials\CredentialsInterface object. (Client error response [status code] 404 [reason phrase] Not Found [url] http://169.254.169.254/latest/meta-data/iam/security-credentials/)
[ec2-user@ip-172-31-42-123 face]$ sudo php index.php
Error retrieving credentials from the instance profile metadata server. When you are not running inside of Amazon EC2, you must provide your AWS access key ID and secret access key in the "key" and "secret" options when creating a client or provide an instantiated Aws\Common\Credentials\CredentialsInterface object. (Client error response [status code] 404 [reason phrase] Not Found [url] http://169.254.169.254/latest/meta-data/iam/security-credentials/)
[ec2-user@ip-172-31-42-123 face]$ sudo php index.php
Image upload done... Here is the URL: https://myproject1-mounika.s3.us-east-2.amazonaws.com/s.jpg[ec2-user@ip-172-31-42-123 face]$
```

Screenshots needed for EC2 & Rekognition

1. Face Detect success screenshot

```
ec2-user@ip-172-31-42-123:/var/www/html/face
ForecastQueryService      Sms
ForecastService           SnowBall
FraudDetector             Sns
FSx                       Sqs
functions.php             Ssm
GameLift                  SSO
Glacier                   SSOIDC
GlobalAccelerator         StorageGateway
Glue                      StreamRequestPayloadMiddleware.php
Greengrass                Sts
GroundStation             Support
GuardDuty                 Swf
Handler                   Textract
HandlerList.php           TraceMiddleware.php
HasDataTrait.php          TranscribeService
HashingStream.php         Transfer
HashInterface.php         Translate
HasMonitoringEventsTrait.php Waf
Health                    WafRegional
History.php               WAFV2
Iam                       Waiter.php
IdempotencyTokenMiddleware.php WorkDocs
imagebuilder              WorkLink
ImportExport              WorkMail
Inspector                 WorkMailMessageFlow
Iot                       WorkSpaces
IoTClickDevicesService    WrappedHttpHandler.php
IoTClickProjects          XRay
[ec2-user@ip-172-31-42-123 src]$ cd ..
[ec2-user@ip-172-31-42-123 aws-sdk-php]$ cd ..
[ec2-user@ip-172-31-42-123 aws]$ cd ..
[ec2-user@ip-172-31-42-123 vendor]$ cd ..
[ec2-user@ip-172-31-42-123 face]$ ls
composer.json composer.lock index.php s.jpg vendor
[ec2-user@ip-172-31-42-123 face]$ sudo php index.php
Cannot read credentials from /var/www/.aws/credentials
[ec2-user@ip-172-31-42-123 face]$ sudo vim index.php
[ec2-user@ip-172-31-42-123 face]$ sudo php index.php
Image upload done... Here is the URL: https://myproject1-mounika.s3.us-east-2.am
[ec2-user@ip-172-31-42-123 face]$ sudo php index.php
Image upload done... Here is the URL: https://myproject1-mounika.s3.us-east-2.amazonaws.com/s.jpgRekognitionClient is working[ec2-user@ip-172-31-42-123 face]$ sudo vim
[ec2-user@ip-172-31-42-123 face]$ sudo php index.php
Image upload done... Here is the URL: https://myproject1-mounika.s3.us-east-2.amazonaws.com/s.jpgRekognitionClient is workingTotally there are 9 faces[ec2-user@ip-172-31-42-123 face]$
```

The output is: Image upload done. Here is the URL: <https://myproject1-mounika.s3.us-east-2.amazonaws.com/s.jpgRekognitionClient> is workingTotally there are 9 faces.