

Dashboards



Root user sign in !

Email: chaitanyapawar51@gmail.com

Password [Forgot password?](#)

[Sign in](#)

[Sign in to a different account](#)

[Create a new AWS account](#)



About Amazon.com Sign In

Amazon Web Services uses information from your Amazon.com account to identify you and allow access to Amazon Web Services. Your use of this site is governed by our Terms of Use and Privacy Policy linked below. Your use of Amazon Web Services products and services is governed by the AWS Customer Agreement linked below unless you have entered into a separate agreement with Amazon Web Services or an AWS Value Added Reseller to purchase these products and services. The AWS Customer Agreement was updated on March 31, 2017. For more information about these updates, see Recent Changes.

1.AWS login Screen with username

The screenshot shows the EC2 Dashboard in the AWS Management Console. The left sidebar includes sections for New EC2 Experience, EC2 Dashboard (New), Events, Tags, Reports, Limits, Instances, Images, and Elastic Block Store. The main content area displays a summary of resources: 1 Running instance, 0 Dedicated Hosts, 1 Volumes, 2 Key pairs, and 0 Placement groups. It also features a "Launch instance" button and a "Service health" section. On the right, there are "Account attributes" and "Explore AWS" sections.

2.EC2 dashboard

AWS Services Resource Groups

Chaitanya Sunil Pawar Global Support

Amazon S3

Buckets

- Batch operations
- Access analyzer for S3

Block public access (account settings)

Feature spotlight

Feedback English (US)

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

Name	Region	Access	Bucket created
aws-web-lax			2020-04-02T05:06:42.000Z

3.S3 dashboard

AWS Services Resource Groups

Chaitanya Sunil Pawar Ohio Support

Amazon Rekognition

Custom Labels

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

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

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

4.Rekognition dashboard

EC2

The screenshot shows the 'Choose an AMI' step in the AWS EC2 wizard. The top navigation bar includes 'Services', 'Resource Groups', and user information ('Chaitanya Sunil Pawar', 'Ohio', 'Support'). Below the navigation is a breadcrumb trail: 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage, 5. Add Tags, 6. Configure Security Group, 7. Review. A 'Cancel and Exit' button is on the right.

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.

A search bar at the top says 'Search for an AMI by entering a search term e.g. "Windows"'. Below it is a 'Quick Start' sidebar with categories: My AMIs, AWS Marketplace, Community AMIs, and a 'Free tier only' section.

The main area lists three AMIs:

- Amazon Linux 2 AMI (HVM), SSD Volume Type** - ami-0e01ce4ee18447327 (64-bit x86) / ami-03201f374ab66a26e (64-bit Arm)
Amazon Linux 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 button (radio buttons for 64-bit (x86) and 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 button (radio buttons for 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 button (radio buttons for 64-bit (x86) and 64-bit (Arm))

At the bottom, there are 'Feedback', 'English (US)', and links to 'Privacy Policy' and 'Terms of Use'.

1. Choosing an AMI

The screenshot shows the 'Choose Instance Type' step in the AWS EC2 wizard. The top navigation bar and breadcrumb trail are identical to the previous screenshot. A 'Filter by:' dropdown set to 'All instance types' and a 'Current generation' dropdown are visible.

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.

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

Buttons at the bottom include 'Cancel', 'Previous', 'Review and Launch' (highlighted in blue), and 'Next: Configure Instance Details'.

2. Choosing an instance type

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.

Review and Launch

3.Adding Storage

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

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.

Review and Launch

4.configur security group

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

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: aws-web-she

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**

5.key pair download

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

Bundle Tasks

Launch Instance

Putty Key Generator

File Key Conversions Help

Key

```
ssh-rsa AAAAB3NzaC1yc2EAAQABAAAQCeuiDlq0LgjTRXUbG7HUX269pbnaNA0argspEc0yyGrIDyPievP-0f7WysysuS1D6jol+UyTtqexZfH3zJoxSW2CPgf4p41+cEU EgHCEZMlyAmlilUO4oHlx5NctzpjmRlsLj38JyjpwDZoDcJMh
```

Key fingerprint: ssh-rsa 2048 73:68:b4:26:1e:88:ad:c7:51:9fc:f5:9b:c7:99

Key comment: imported-openssl-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

Instances

Instance ID	Description	Status Checks	Monitoring
i-040a8d29405b44a32	Putty	Passing	Not Enabled

Instance ID: i-040a8d29405b44a32

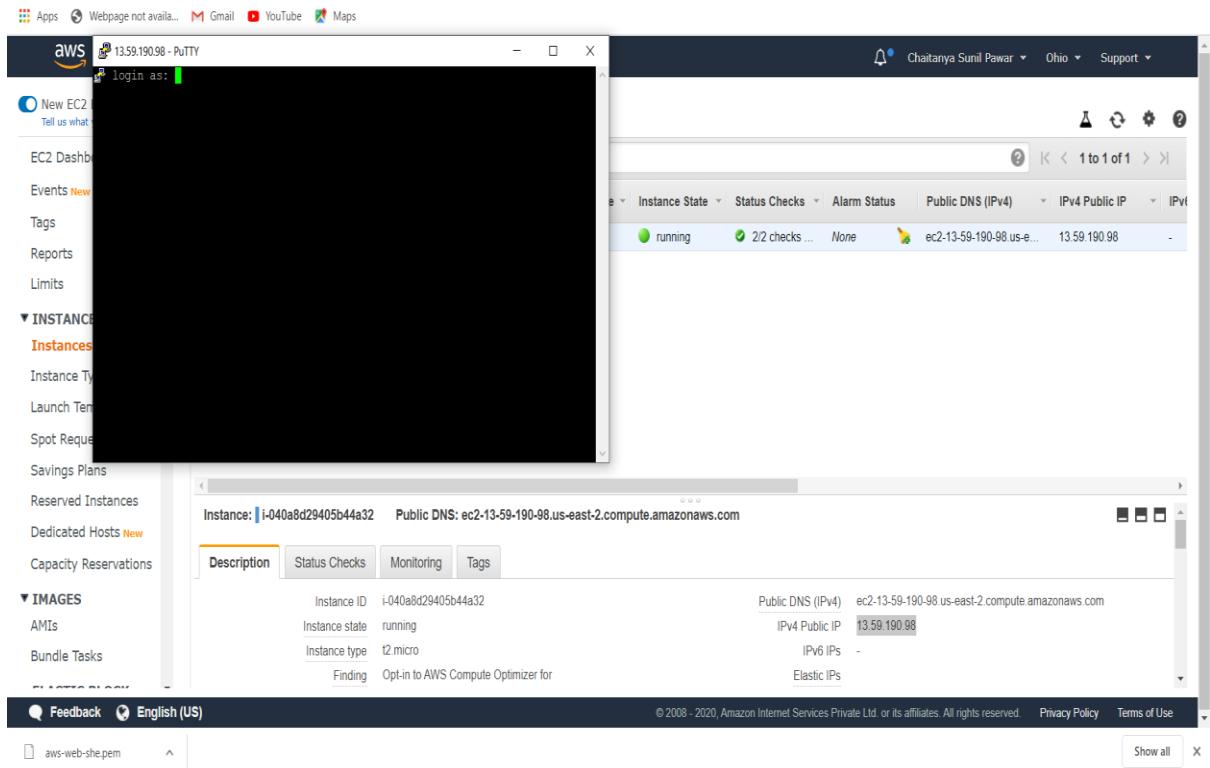
Instance state: running

Instance type: t2.micro

Finding: Opt-in to AWS Compute Optimizer for

PUTTYgen conversion from pem to ppk

6.PUTTYgen conversion from pem to ppk



7.Logged in EC2 black screen

S3

Create bucket

General configuration

Bucket name: aws-web-la

Region: US East (Ohio) us-east-2

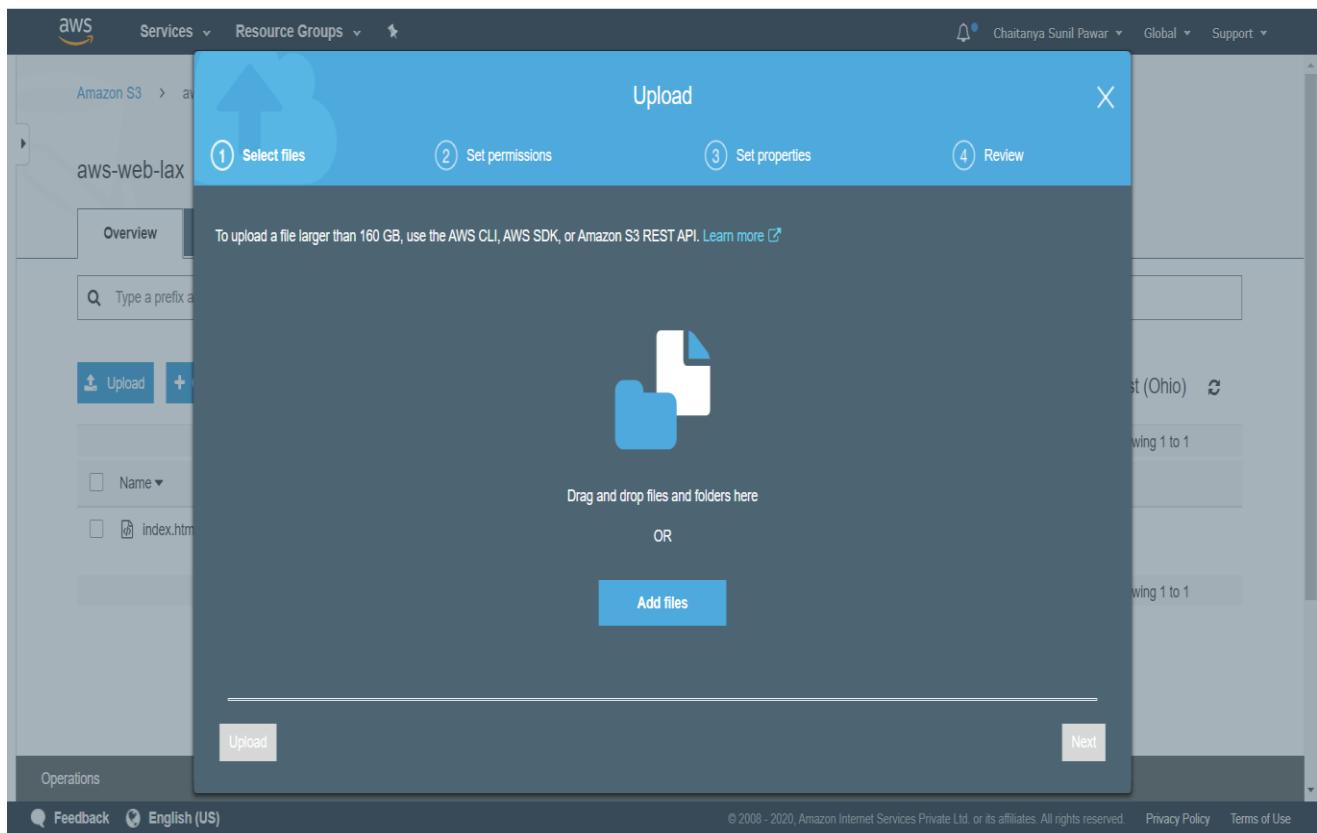
Bucket settings for Block Public Access

Block all public access:

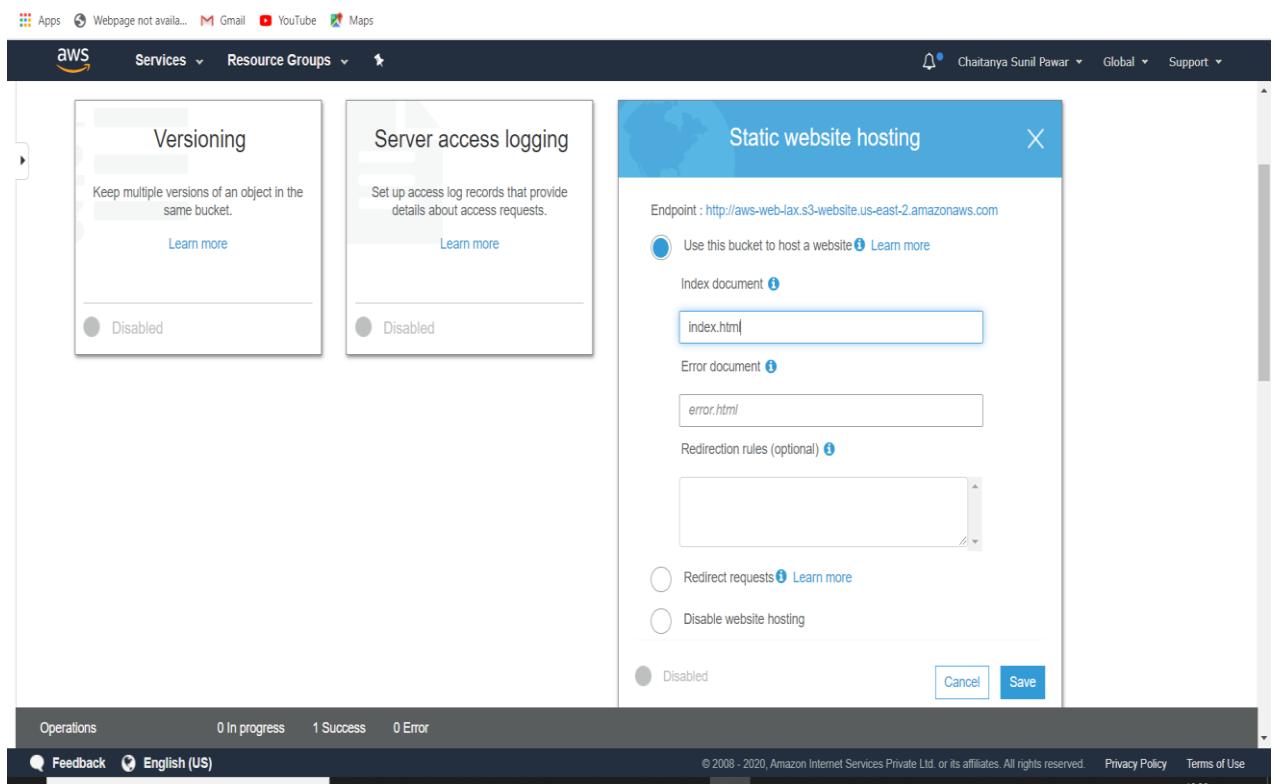
Block public access to buckets and objects granted through new access control lists (ACLs):

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

1.Creating a bucket



2.Uploading a Object



3.Enabling Static website

The screenshot shows the AWS S3 Properties page for an object named 'index.html' in the 'aws-web-lax' bucket. The 'Properties' tab is selected. Key details shown include:

- Owner:** a3c485b82eb390076804bfa894b3402d5b6a416b45577aab05240336d033819e
- Last modified:** Apr 2, 2020 10:37:25 AM GMT+0530
- Etag:** bc1af6d500f20129599685f3859cdff4
- Storage class:** Standard
- Server-side encryption:** None

Below the properties, there's a summary of operations: 0 In progress, 3 Success, 0 Error. At the bottom, there are links for Feedback, English (US), Privacy Policy, and Terms of Use.

4. Making the object Public

The screenshot shows a browser window displaying the content of the 'index.html' file from the previous step. The URL is https://aws-web-lax.s3.us-east-2.amazonaws.com/index.html. The page content is: "Hello all, I am chaitanya, I wish all are safe".

5. Checking S3 link in browser

Rekognition

The screenshot shows the AWS Rekognition service interface for 'Facial analysis'. On the left sidebar, under 'Demos', 'Facial analysis' is selected. The main content area displays a woman driving a yellow car, with a white bounding box highlighting her face. Below the image are two buttons: 'Choose a sample image' and 'Use your own Image'. The right side shows a results table with the following data:

Result	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 %

[Show more](#)

1. Face Detect

The screenshot shows the AWS Rekognition service interface for 'Face comparison'. On the left sidebar, under 'Demos', 'Face comparison' is selected. The main content area displays two images: a reference face of a girl in a pink shirt and a comparison face of three girls. Below each image is a 'Choose a sample image' button. The right side shows a results table with the following data:

Comparison	Similarity (%)
= (two girls)	99.8 %
≠ (girl vs. girl)	
≠ (girl vs. two girls)	

2. Face Compare

AWS Services Resource Groups Chaitanya Sunil Pawar Ohio Support

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

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.

3.Celebrity rekognition

AWS Services Resource Groups Chaitanya Sunil Pawar Ohio Support

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

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.

4.Text in Image

EC2 & S3

The screenshot shows the AWS Management Console interface. On the left, the navigation pane includes links for New EC2 Experience, EC2 Dashboard, Events, Tags, Reports, Limits, Instances, 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 has two tabs: EC2 Instances and S3 Services. The EC2 Instances tab is active, displaying a list of instances. One instance is selected, showing its details: Instance ID i-040a8d29405b44a32, Public DNS ec2-13-59-190-98.us-east-2.compute.amazonaws.com, Instance state running, Instance type t2.micro, and a note about Opt-in to AWS Compute Optimizer. The S3 Services tab shows a list of buckets: Chaitanya-Sunil-Pawar, Ohio, and Support. A large central window displays a terminal session on the selected EC2 instance (ip-172-31-18-37). The terminal output shows the user performing several commands:

```
Setting up swap space version 1, size = 1024 MiB (1073737728 bytes)
no label, UUID=452b88c3-1ac0-485c-8c40-694465c12c37
[ec2-user@ip-172-31-18-37 ~]$ 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=452b88c3-1ac0-485c-8c40-694465c12c37
[ec2-user@ip-172-31-18-37 ~]$ sudo /sbin/swapon /var/swap.1
swapon: /var/swap.1: insecure permissions 0644, 0600 suggested.
[ec2-user@ip-172-31-18-37 ~]$ sudo php -d memory_limit=-1 ~/composer.phar require aws/aws-sdk-php
Using version ^2.0 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 (Bases 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-172-31-18-37 ~]$
```

1. Installing aws-sdk

Screenshot of the AWS Management Console showing the EC2 Dashboard. A terminal window is open on an Amazon Linux 2 AMI instance, displaying the command output for installing PHP. The terminal shows the package manager updating dependencies and installing PHP 5.4.16-46.amzn2.0.2.

```

login as: ec2-user
Authenticating with public key "imported-openssh-key"
[ec2-user@ip-172-31-18-37 ~]$ sudo yum update
[ec2-user@ip-172-31-18-37 ~]$ sudo yum install php
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                                         | 2.4 kB   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: ph
p-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

```

The instance details are shown below:

Instance ID	i-040a8d29405b44a32	Public DNS	ec2-13-59-190-98.us-east-2.compute.amazonaws.com
Instance state	running	IPv4 Public IP	13.59.190.98
Instance type	t2.micro	IPv6 IPs	-
Finding	Opt-in to AWS Compute Optimizer for recommendations.	Elastic IPs	-

Terminal Output:

```

login as: ec2-user
Authenticating with public key "imported-openssh-key"

[ec2-user@ip-172-31-18-37 ~]$ sudo yum update
[ec2-user@ip-172-31-18-37 ~]$ sudo yum install php
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                                         | 2.4 kB   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: ph
p-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: ht
tpd-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_6
4
--> Processing Dependency: httpd-filesystem for package: httpd-2.4.41-1.amzn2.0.
1.x86_64

```

2. Installing php

```

cd /var/www/html
sudo mkdir face
cd face
sudo php -d memory_limit=-1 ~/composer.phar require aws/aws-sdk-php

In case if you get memory error -
    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 = 'aws-web-lax';
$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;
}

```

3.index.php file code

```

no label, UUID=452b88c3-lac0-485c-8c40-6944b5c12c37
[ec2-user@ip-172-31-18-37 face]$ sudo /sbin/swapon /var/swap.1
swapon: /var/swap.1: insecure permissions 0644, 0600 suggested.
[ec2-user@ip-172-31-18-37 face]$ sudo php -d memory_limit=-1 ~/composer.phar require aws/aws-sdk-php
Using version ^2.0 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-172-31-18-37 face]$ sudo wget https://i.pinimg.com/originals/b9/7e/a3/b97ea33b5842c7894b804923c6c05580.jpg
--2020-04-02 05:21:21-- https://i.pinimg.com/originals/b9/7e/a3/b97ea33b5842c7894b804923c6c05580.jpg
Resolving i.pinimg.com (i.pinimg.com)... 23.197.108.230, 2a04:4e42:3c::84
Connecting to i.pinimg.com (i.pinimg.com)|23.197.108.230|: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

2020-04-02 05:21:21 (5.61 MB/s) - 'b97ea33b5842c7894b804923c6c05580.jpg' saved [215551/215551]

[ec2-user@ip-172-31-18-37 face]$ ls
b97ea33b5842c7894b804923c6c05580.jpg composer.json composer.lock vendor
[ec2-user@ip-172-31-18-37 face]$ sudo mv b97ea33b5842c7894b804923c6c05580.jpg sample.jpg
[ec2-user@ip-172-31-18-37 face]$ ls
composer.json composer.lock sample.jpg vendor
[ec2-user@ip-172-31-18-37 face]$ sudo vim index.php
[ec2-user@ip-172-31-18-37 face]$ sudo php index.php
You must specify a non-null value for the Body or SourceFile parameters.
[ec2-user@ip-172-31-18-37 face]$ sudo vim index.php
[ec2-user@ip-172-31-18-37 face]$ sudo php index.php
PHP Parse error: syntax error, unexpected '<' in /var/www/html/face/index.php on line 55
[ec2-user@ip-172-31-18-37 face]$ sudo php index.php
PHP Parse error: syntax error, unexpected '<' in /var/www/html/face/index.php on line 55
[ec2-user@ip-172-31-18-37 face]$ sudo vim index.php
[ec2-user@ip-172-31-18-37 face]$ sudo php index.php
Image upload done... Here is the URL: https://aws-web-lax.s3.us-east-2.amazonaws.com/sample.jpg[ec2-user@ip-172-31-18-37 face]$ 

```

```

HTTP request sent, awaiting response... 200 OK
Length: 215551 (210K) [image/jpeg]
Saving to: 'b97ea33b5842c7894b804923c6c05580.jpg'

100%[=====]

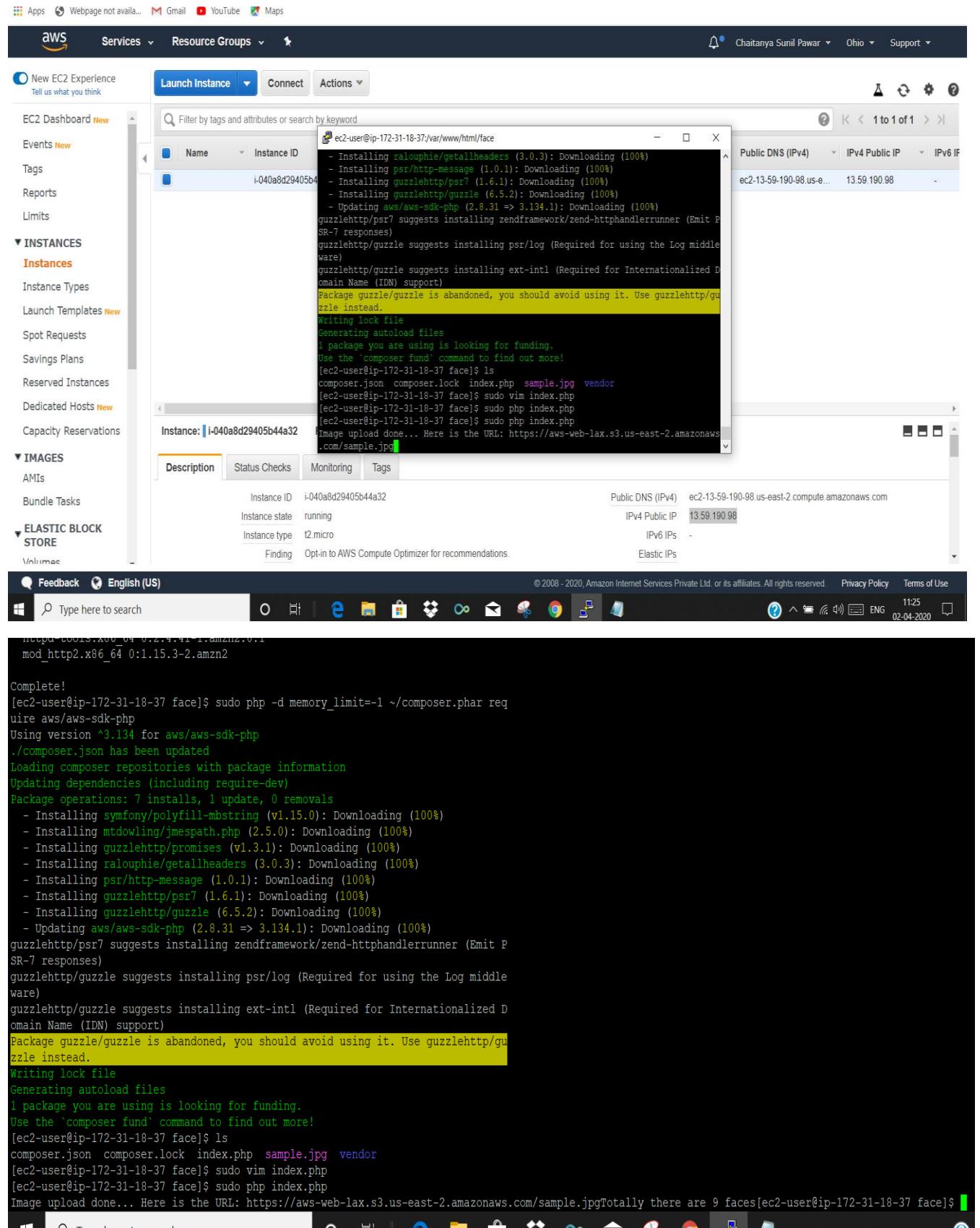
2020-04-02 05:21:21 (5.61 MB/s) - 'b97ea33b5842c7894b804923c6c05580.jpg' saved [215551/215551]

[ec2-user@ip-172-31-18-37 face]$ ls
b97ea33b5842c7894b804923c6c05580.jpg composer.json composer.lock vendor
[ec2-user@ip-172-31-18-37 face]$ sudo mv b97ea33b5842c7894b804923c6c05580.jpg sample.jpg
[ec2-user@ip-172-31-18-37 face]$ ls
composer.json composer.lock sample.jpg vendor
[ec2-user@ip-172-31-18-37 face]$ sudo vim index.php
[ec2-user@ip-172-31-18-37 face]$ sudo php index.php
You must specify a non-null value for the Body or SourceFile parameters.
[ec2-user@ip-172-31-18-37 face]$ sudo vim index.php
[ec2-user@ip-172-31-18-37 face]$ sudo php index.php
PHP Parse error: syntax error, unexpected '<' in /var/www/html/face/index.php on line 55
[ec2-user@ip-172-31-18-37 face]$ sudo php index.php
PHP Parse error: syntax error, unexpected '<' in /var/www/html/face/index.php on line 55
[ec2-user@ip-172-31-18-37 face]$ sudo vim index.php
[ec2-user@ip-172-31-18-37 face]$ sudo php index.php
Image upload done... Here is the URL: https://aws-web-lax.s3.us-east-2.amazonaws.com/sample.jpg[ec2-user@ip-172-31-18-37 face]$ 

```

4.Upload success screenshot

Screenshot needed for Ec2 & Rekognition



1. Face detection success screenshot