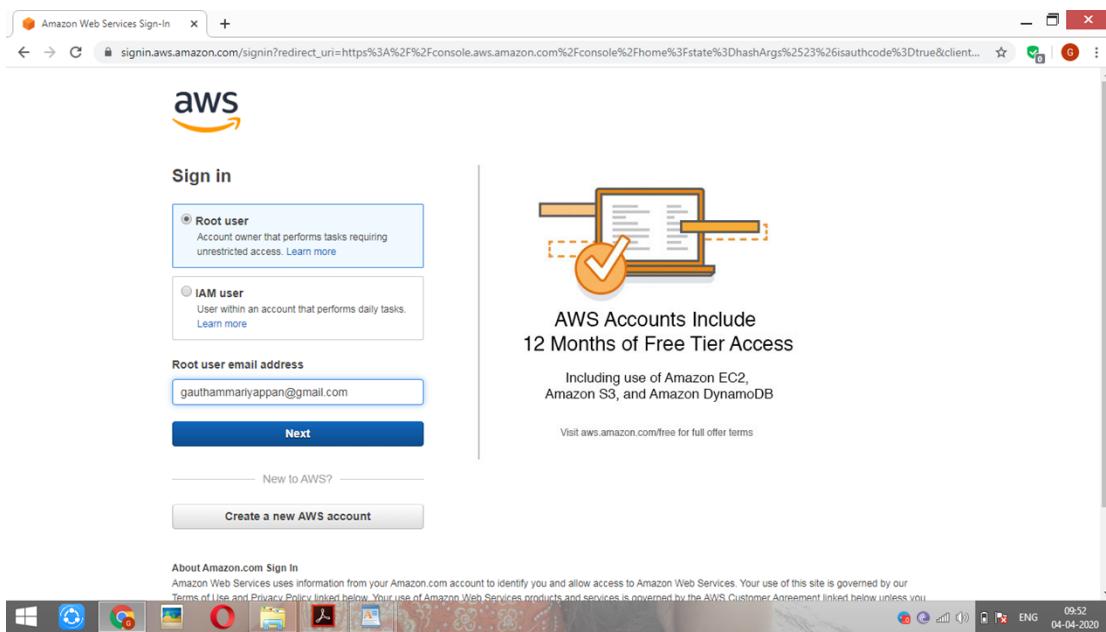
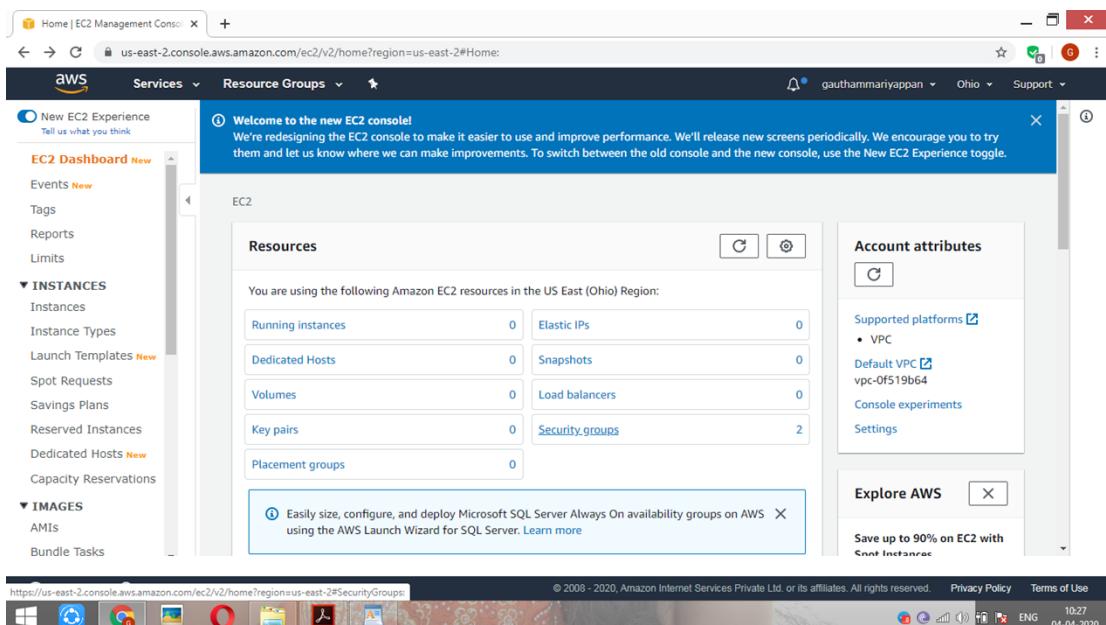


# AWS(Dashboard)

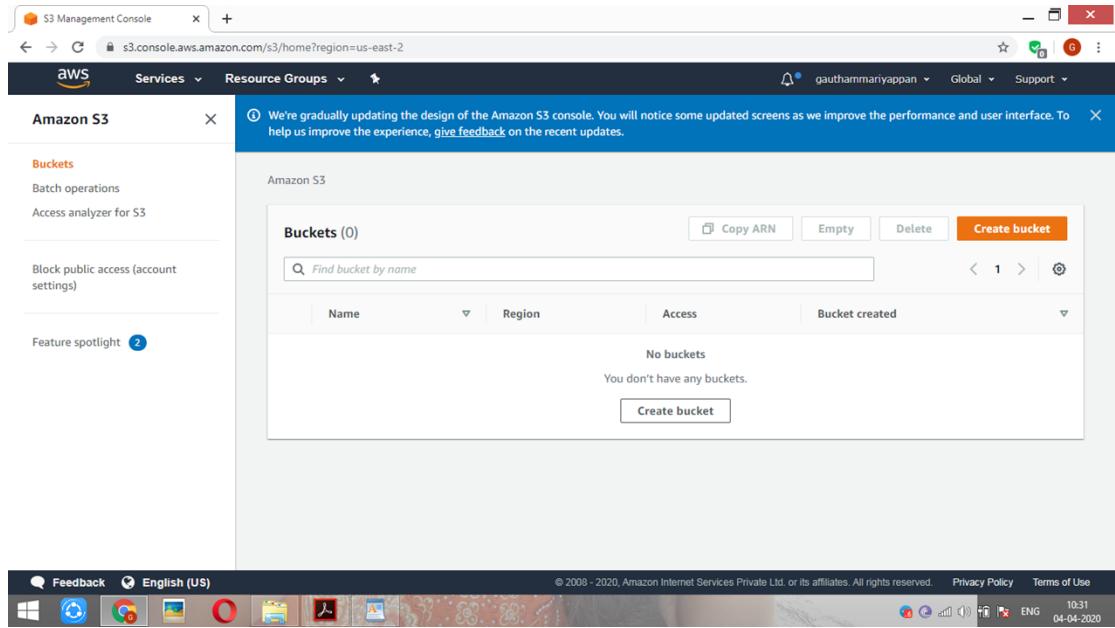
## 1: AWS login screen with username



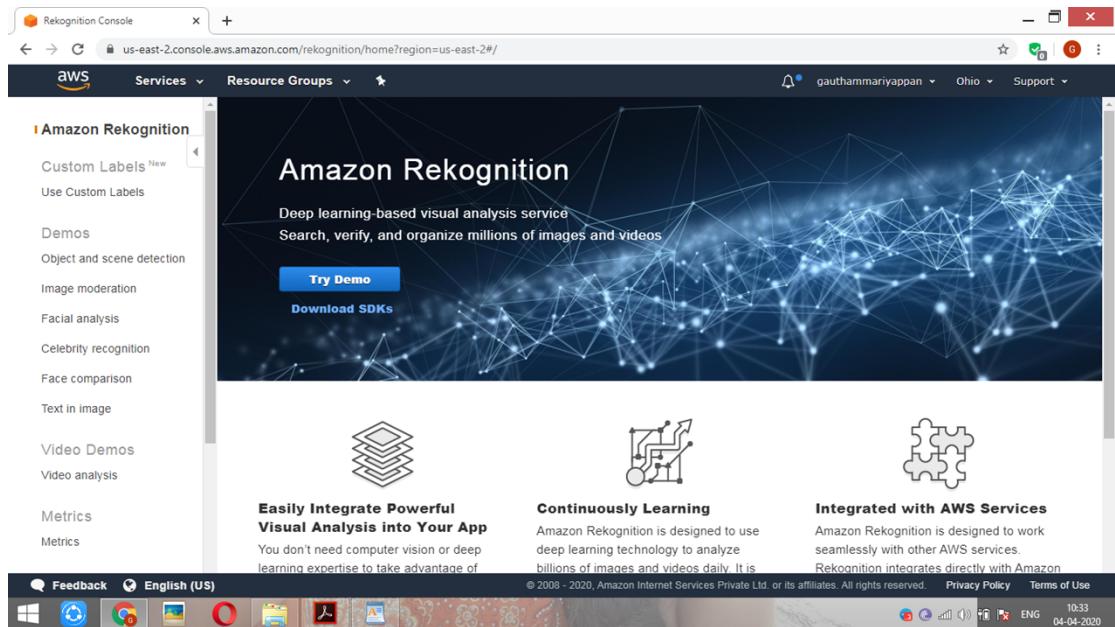
## 2: EC2 Dashboard



## 3: S3 Dashboard



## 4: Rekognition Dashboard



## 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

	Name	Description	Root device type	Virtualization type	ENAv Enabled	Select
	<a href="#">Amazon Linux 2 AMI (HVM), SSD Volume Type</a>	- ami-0e01ce4ee18447327 (64-bit x86) / ami-03201f374ab66a26e (64-bit Arm)			Yes	<input checked="" type="button"/>
	<a href="#">Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type</a>	- ami-01b01bbd08f24c7a8			Yes	<input type="button"/>
	<a href="#">Red Hat Enterprise Linux 8 (HVM), SSD Volume Type</a>	- ami-0520e698dd500b1d1 (64-bit x86) / ami-0520e698dd500b1d1 (64-bit Arm)			Yes	<input type="button"/>

## 2: Choosing an Instance type

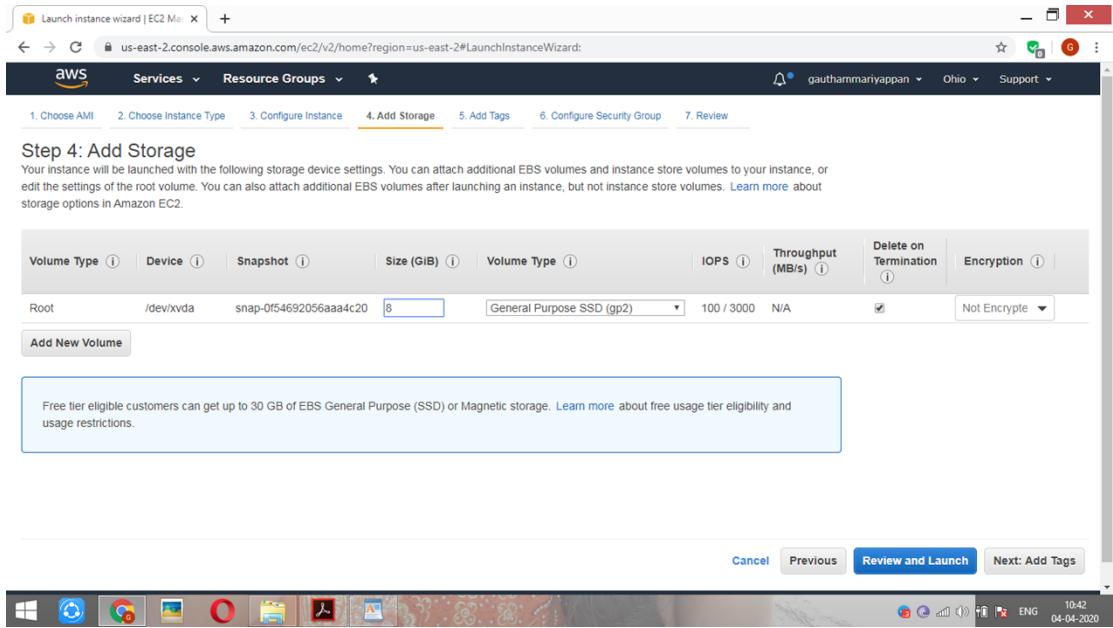
Step 2: Choose an Instance Type

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.

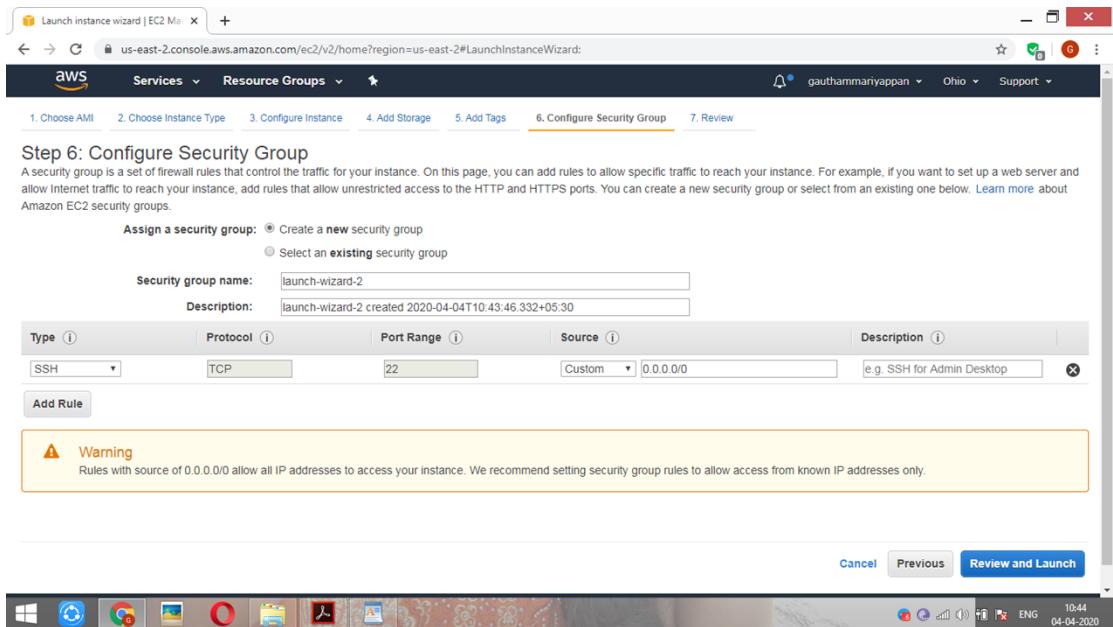
Filter by:	All instance types	Current generation	Show/Hide Columns				
Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)							
	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate
<input checked="" type="checkbox"/>	General purpose	<a href="#">t2.micro</a> Free tier eligible	1	1	EBS only	-	Low to Moderate
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate
<input type="checkbox"/>	General purpose	t2.large	2	8	EBS only	-	Low to Moderate
<input type="checkbox"/>	General purpose	t2.xlarge	4	16	EBS only	-	Moderate

Cancel Previous Review and Launch Next: Configure Instance Details

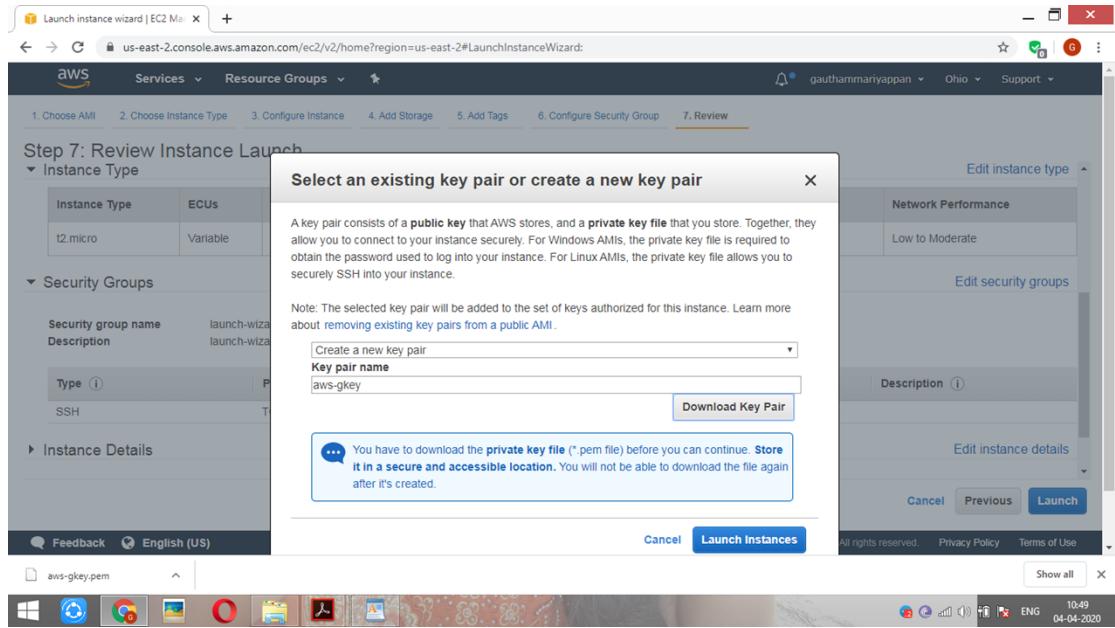
## 3: Adding Storage



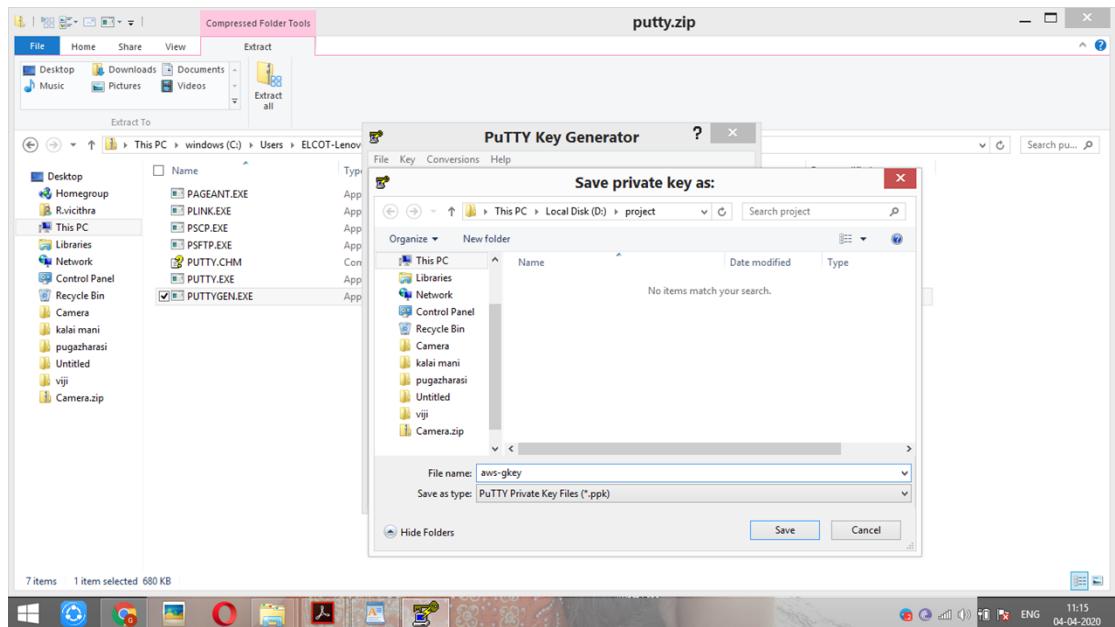
## 4: Configuring Security group



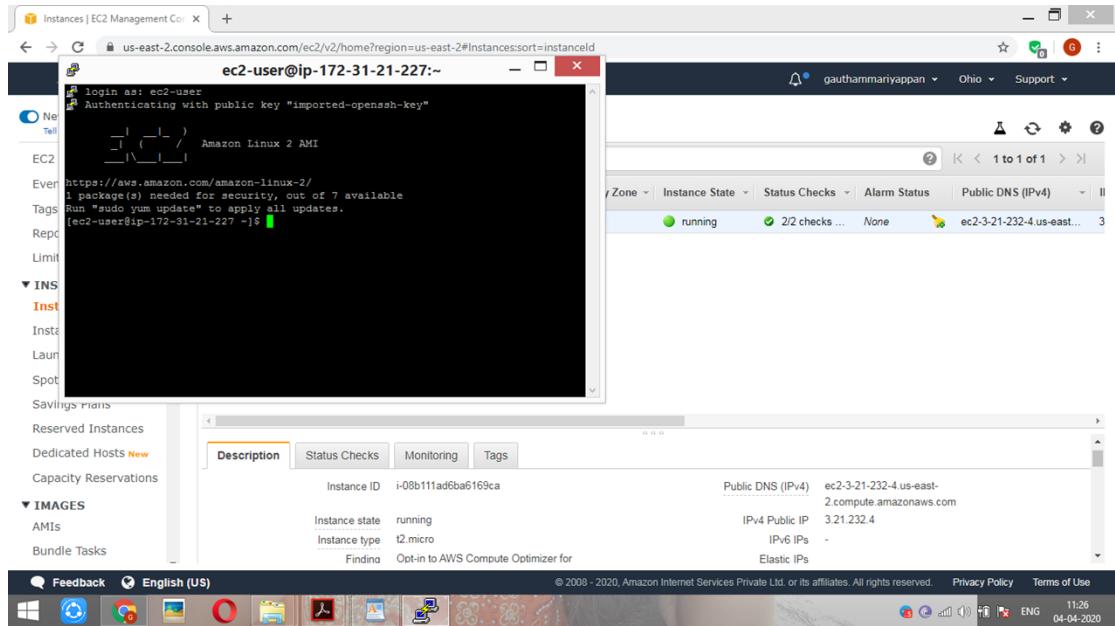
## 5: Key Pair Download



## 6: puTTYgen conversion from PEM to PPK

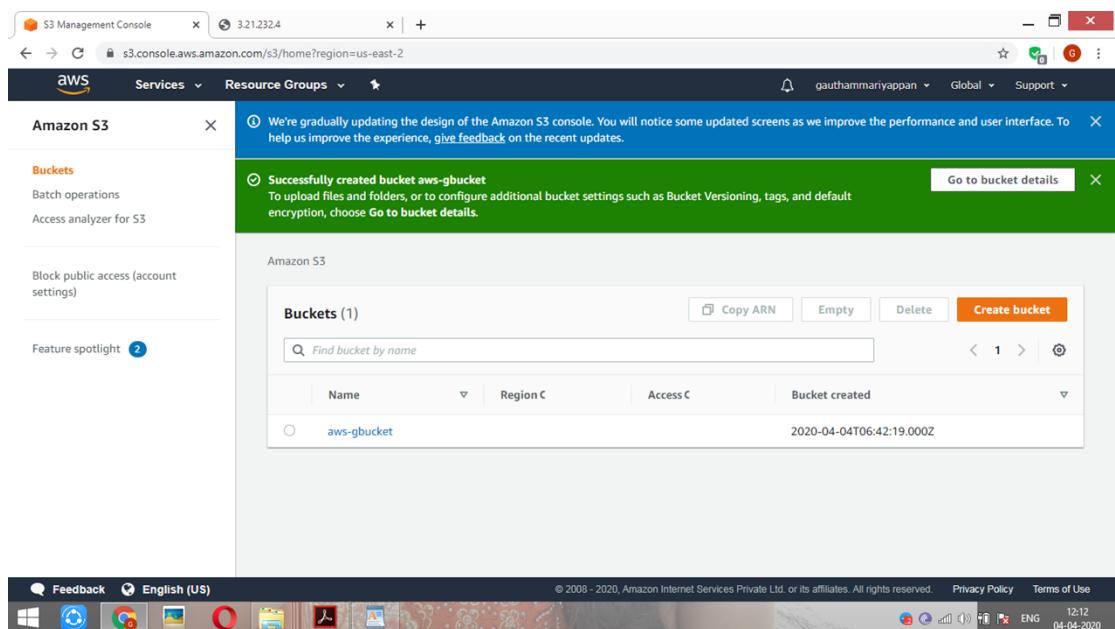


## 7: Logged in EC2 BlackScreen



## S3

### 1: Creating a Bucket



### 2: Uploading an Object

The screenshot shows the AWS S3 Management Console interface. At the top, the URL is <https://s3.console.aws.amazon.com/s3/buckets/aws-gbucket/?region=us-east-2>. The navigation bar includes 'Services', 'Resource Groups', and 'Support'. The user 'gauthammaryappan' is logged in. The main content area shows the 'aws-gbucket' bucket. The 'Overview' tab is selected, followed by 'Properties', 'Permissions', 'Management', and 'Access points'. A search bar at the top of the content area contains the placeholder 'Type a prefix and press Enter to search. Press ESC to clear.' Below the search bar are buttons for 'Upload', '+ Create folder', 'Download', and 'Actions'. The region is set to 'US East (Ohio)'. The table below lists one object: 'demo.html'. The object was last modified on Apr 4, 2020, at 12:24:50 PM GMT+0530, has a size of 46.0 B, and is stored in the Standard storage class. The status bar at the bottom shows 'Viewing 1 to 1'.

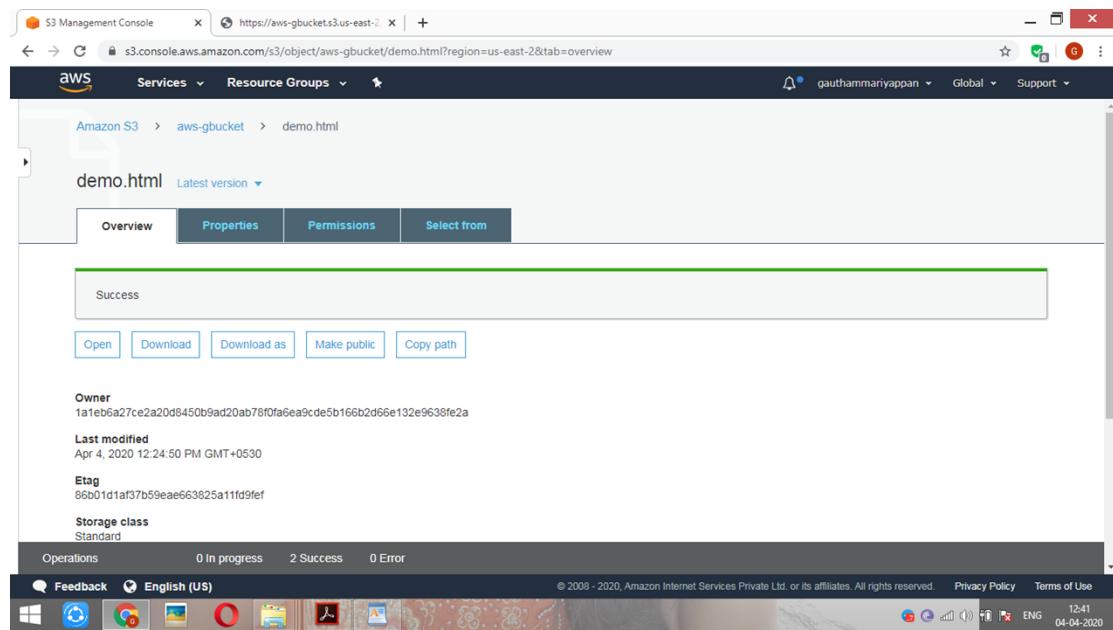
### 3: Enabling Static Website

The screenshot shows the 'Properties' tab of the AWS S3 Management Console for the 'aws-gbucket' bucket. The URL is <https://aws-gbuckets.s3.us-east-2.amazonaws.com/>. The page displays several configuration options:

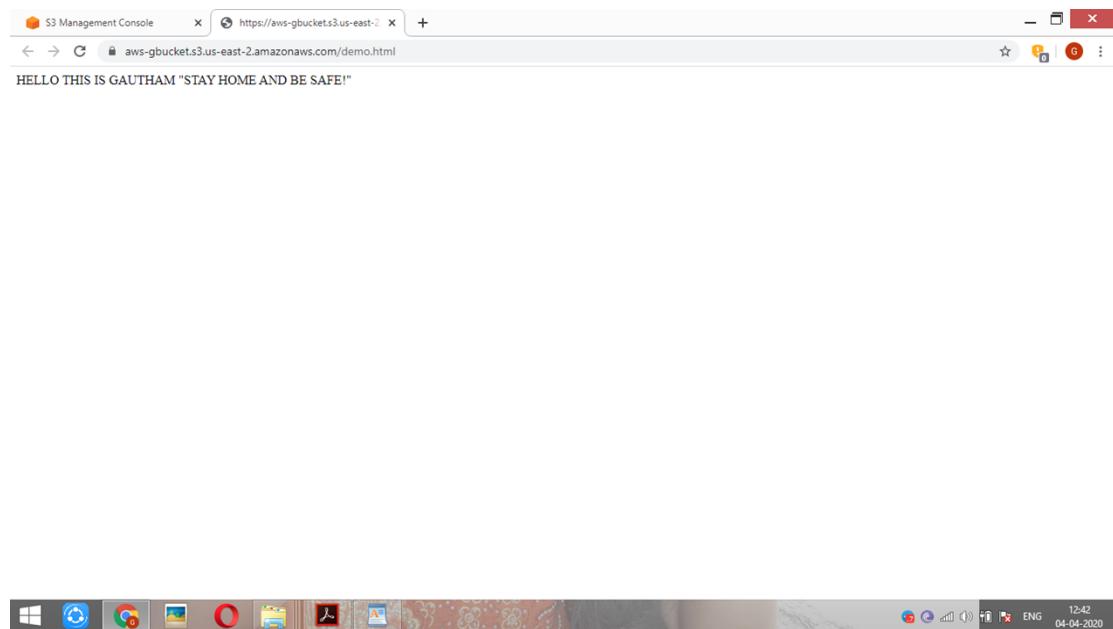
- Versioning**: Keep multiple versions of an object in the same bucket. Status: Disabled.
- Server access logging**: Set up access log records that provide details about access requests. Status: Disabled.
- Static website hosting**: Host a static website, which does not require server-side technologies. Status: Bucket hosting (indicated by a checked checkbox).
- Object-level logging**: Record object-level API activity using the CloudTrail data events feature (additional cost). Status: Disabled.
- Default encryption**: Automatically encrypt objects when stored in Amazon S3. Status: Disabled.

The status bar at the bottom shows 'Viewing 1 to 1'.

### 4: Making the Object Public



## 5: Checking the S3 link on the browser



## Rekognition

## 1: Face Detect

The screenshot shows the 'Facial analysis' demo page. On the left, a sidebar lists various features: Custom Labels, Demos, Object and scene detection, Image moderation, Facial analysis (which is selected and highlighted in orange), Celebrity recognition, Face comparison, Text in image, Video Demos, Video analysis, Metrics, and Metrics. The main content area displays a portrait of a man with a beard and mustache. A blue bounding box highlights his face, and several small dots are placed on his forehead, nose, and mouth to indicate detected landmarks. Below the image, there's a link to 'Choose a sample image' and another to 'Use your own image'. To the right, a 'Results' section shows the following data:

Attribute	Value	Confidence (%)
looks like a face	99.9 %	
appears to be male	99.7 %	
age range	31 - 47 years old	
not smiling	85.4 %	
appears to be calm	91 %	
not wearing glasses	99.2 %	

At the bottom right, there's a 'Done with the demo?' link with a 'Learn more' button. The footer includes standard links like Feedback, English (US), Privacy Policy, Terms of Use, and a timestamp of 12:51 on 04-04-2020.

## 2: Face Compare

The screenshot shows the 'Face comparison' demo page. The sidebar is identical to the previous one, with 'Facial analysis' selected. The main area displays three sample images of men: Christian Bale, Jim Carrey, and Steve Jobs. Below these are two sections for 'Choose a sample image' and 'Use your own image'. To the right, a 'Results' section compares pairs of faces and shows their similarity percentages:

Comparison	Similarity (%)
Christian Bale vs. Christian Bale	97.3 %
Jim Carrey vs. Jim Carrey	96.1 %
Steve Jobs vs. Steve Jobs	97.3 %

At the bottom right, there's a 'Request' button. The footer includes standard links like Feedback, English (US), Privacy Policy, Terms of Use, and a timestamp of 12:59 on 04-04-2020.

## 3: Celebrity Recognition

The screenshot shows the Amazon Rekognition console interface. On the left, a sidebar lists various services: Custom Labels, Demos, Celebrity recognition (which is selected), and Video Demos. The main area displays a photograph of Robert Downey Jr. in a suit, with a blue bounding box highlighting his face. To the right, the results panel shows a small thumbnail of the detected face, the name "Robert Downey Jr.", and a match confidence of "100 %". Below this are sections for "Request" and "Response". At the bottom, there are standard browser controls and a taskbar.

## 4: Text in image

The screenshot shows the Amazon Rekognition console interface for "Text in image" detection. The sidebar includes options like Custom Labels, Demos, and Text in image (selected). The main area features a movie scene from "The Shawshank Redemption" with overlaid text: "HOPE IS A DANGEROUS THING HOPE CAN DRIVE A MAN INSANE". The results panel on the right shows the detected text segments: "HOPE | IS | A | DANGEROUS |", "THING | HOPE | CAN |", "DRIVE | A | MAN | INSANE. |", "ELLIS | EDOG |", and "MAGICALQUOTE |". There are also sections for "Request" and "Response". The bottom of the screen shows a taskbar and system status.

## EC2&S3

### 1: Installing AWS-sdk

```
[ec2-user@ip-172-31-21-227 face]$ sudo php -d memory_limit=-1 ~/composer.phar require aws/aws-sdk-php
Using version "3" 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 aws/aws-sdk (Adds support for creating jobs in AWS Import/Export)
aws/aws-sdk-php suggests installing aws/aws-sdk-ec2-apc (Allows service description options 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-172-31-21-227 face]$
```

## 2: Installing php

```
ec2-user@ip-172-31-21-227:~#
=====
Installing:
php                           x86_64           5.4.16-46.amzn2.0.2          amzn2-core           1.4 M
Installing for dependencies:
libzip010-compat              x86_64           0.10.1-9.amzn2.0.5          amzn2-core           30 k
php-cli                         x86_64           5.4.16-46.amzn2.0.2          amzn2-core           2.8 M
php-common                      x86_64           5.4.16-46.amzn2.0.2          amzn2-core           563 k
=====
Transaction Summary
=====
Install 1 Package (+3 Dependent packages)

Total download size: 4.7 M
Installed size: 17 M
Is this ok [y/d/N]: y
Downloading packages:
(1/4): libzip010-compat-0.10.1-9.amzn2.0.5.x86_64.rpm | 30 kB 00:00:00
(2/4): php-5.4.16-46.amzn2.0.2.x86_64.rpm             | 1.4 MB 00:00:00
(3/4): php-common-5.4.16-46.amzn2.0.2.x86_64.rpm       | 563 kB 00:00:00
(4/4): php-cli-5.4.16-46.amzn2.0.2.x86_64.rpm         | 2.8 MB 00:00:00
=====
Total                                         17 MB/s | 4.7 MB 00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : libzip010-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
  Installing : php-5.4.16-46.amzn2.0.2.x86_64        4/4
  Verifying  : libzip010-compat-0.10.1-9.amzn2.0.5.x86_64 1/4
  Verifying  : php-cli-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-5.4.16-46.amzn2.0.2.x86_64         4/4
Installed:
  php.x86_64 0:5.4.16-46.amzn2.0.2
Dependency Installed:
  libzip010-compat.x86_64 0:0.10.1-9.amzn2.0.5          php-cli.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-172-31-21-227 ~]$
```

## 3: Index.php File Code

```

sudo /sbin/swapp /var/swapi
sudo /sbin/swapon /var/swapi

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-gbucket';
$keyname = '.jpg';

$b3 = $S3Client::factory([
    'profile'     => 'default',
    'region'      => 'us-east-2',
    'version'     => '2006-03-01',
    'signature'   => 'v4'
]);

try {
    $result = $b3->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;
}

:wq

```

## 4: Upload Success Screenshot

```

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-172-31-21-227 html]$ sudo mkdir face
mkdir: cannot create directory `face': File exists
[ec2-user@ip-172-31-21-227 html]$ cd face
[ec2-user@ip-172-31-21-227 face]$ pwd
/var/www/html/face
[ec2-user@ip-172-31-21-227 face]$ sudo php -d memory_limit=-1 ~/composer.phar require aws/aws-sdk-php
Using version *.* for aws/aws-sdk-php
./composer.json has been updated
Loading composer repositories with package information
Updating dependencies (including require-dev)
Nothing to install or update
Package guzzle/guzzle is abandoned, you should avoid using it. Use guzzlehttp/guzzle instead.
Writing lock file
[ec2-user@ip-172-31-21-227 face]$ ls
composer.json  composer.lock  index.php  sample.jpg  s.jpg  vendor
[ec2-user@ip-172-31-21-227 face]$ sudo rm index.php
[ec2-user@ip-172-31-21-227 face]$ sudo rm sample.jpg
[ec2-user@ip-172-31-21-227 face]$ sudo rm s.jpg
[ec2-user@ip-172-31-21-227 face]$ ls
composer.json  composer.lock  vendor
[ec2-user@ip-172-31-21-227 face]$ sudo wget https://i.pinimg.com/originals/b9/7e/a3/b97ea33b5842c7894b804923c6c05580.jpg
2020-04-04 10:23:05 -- https://i.pinimg.com/originals/b9/7e/a3/b97ea33b5842c7894b804923c6c05580.jpg
Resolving i.pinimg.com (i.pinimg.com)... 151.101.248.84, 2a04:4e42:2f:84
Connecting to i.pinimg.com (i.pinimg.com)|151.101.248.84|: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

[ec2-user@ip-172-31-21-227 face]$ sudo mv b97ea33b5842c7894b804923c6c05580.jpg sample.jpg
[ec2-user@ip-172-31-21-227 face]$ sudo vim index.php
[ec2-user@ip-172-31-21-227 face]$ sudo php index.php
Image upload done... Here is the URL: https://aws-gbucket.s3.us-east-2.amazonaws.com/sample.jpg[ec2-user@ip-172-31-21-227 face]$ 

```

## EC2&Rekognition

### 1: Face Detect Success Screenshot

```
ec2-user@ip-172-31-21-227:/var/www/html/face
[ec2-user@ip-172-31-21-227 face]$ sudo php -d memory_limit=-1 ~/composer.phar require aws/aws-sdk-php
Using version ^3.134 for aws/aws-sdk-php
./composer.json has been updated
Locking dependencies into package.json with package information
Updating dependencies (including require-dev)
Nothing to install or update
Package guzzle/guzzle is abandoned, you should avoid using it. Use guzzlehttp/guzzle instead.
Generating autoload files
1 package you are using is looking for funding.
Use the 'composer fund' command to find out more!
[ec2-user@ip-172-31-21-227 face]$ sudo php index.php
Image upload done... Here is the URL: https://aws-gbucket.s3.us-east-2.amazonaws.com/s.jpgTotally there are 9 faces[ec2-user@ip-172-31-21-227 face]$
```

