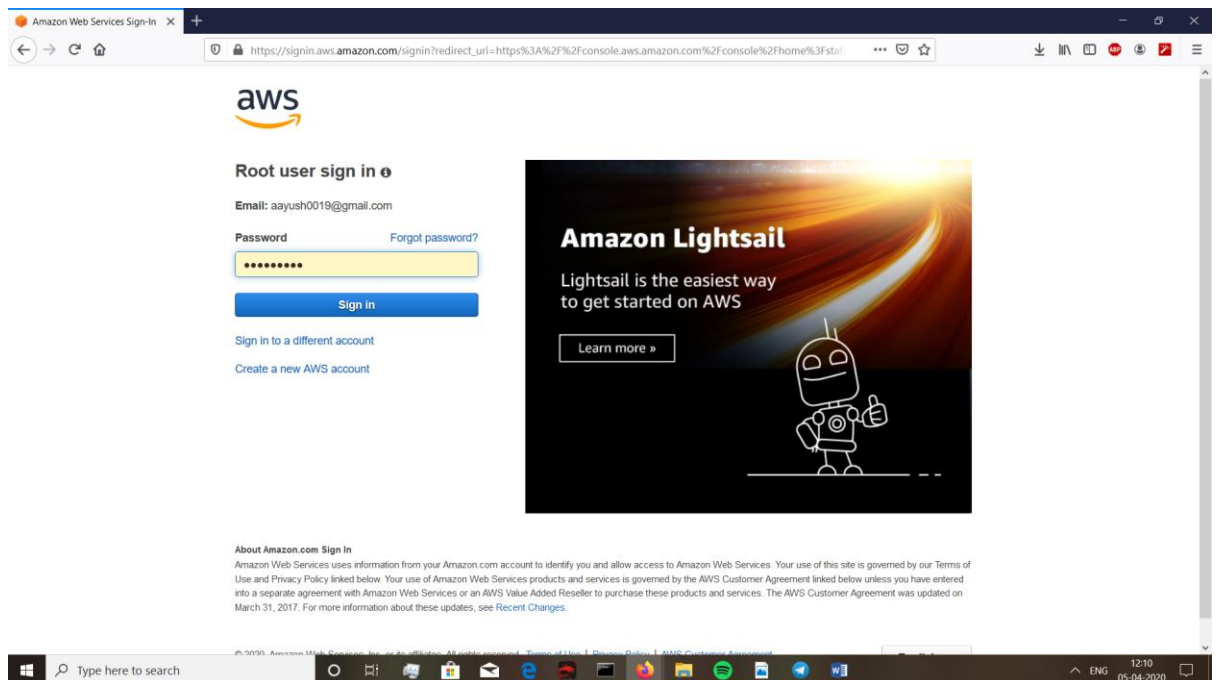


By – Aayush Gupta – aayush0019@gmail.com

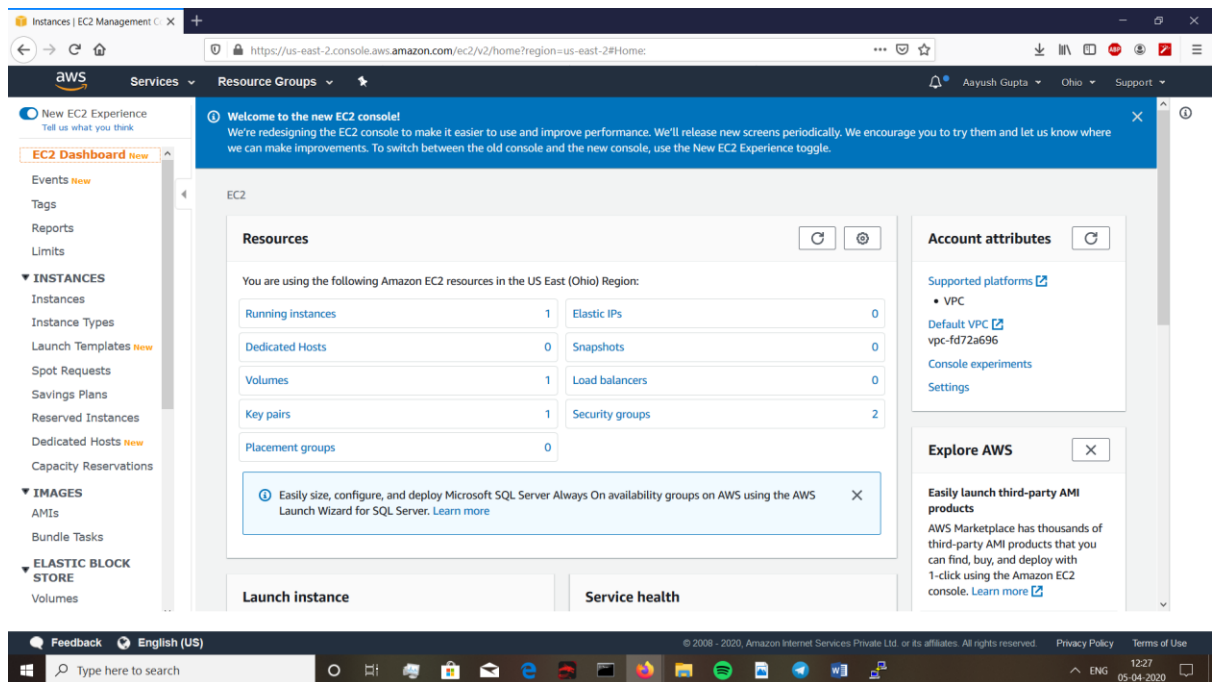
Screenshots

Screenshots needed for Dashboards

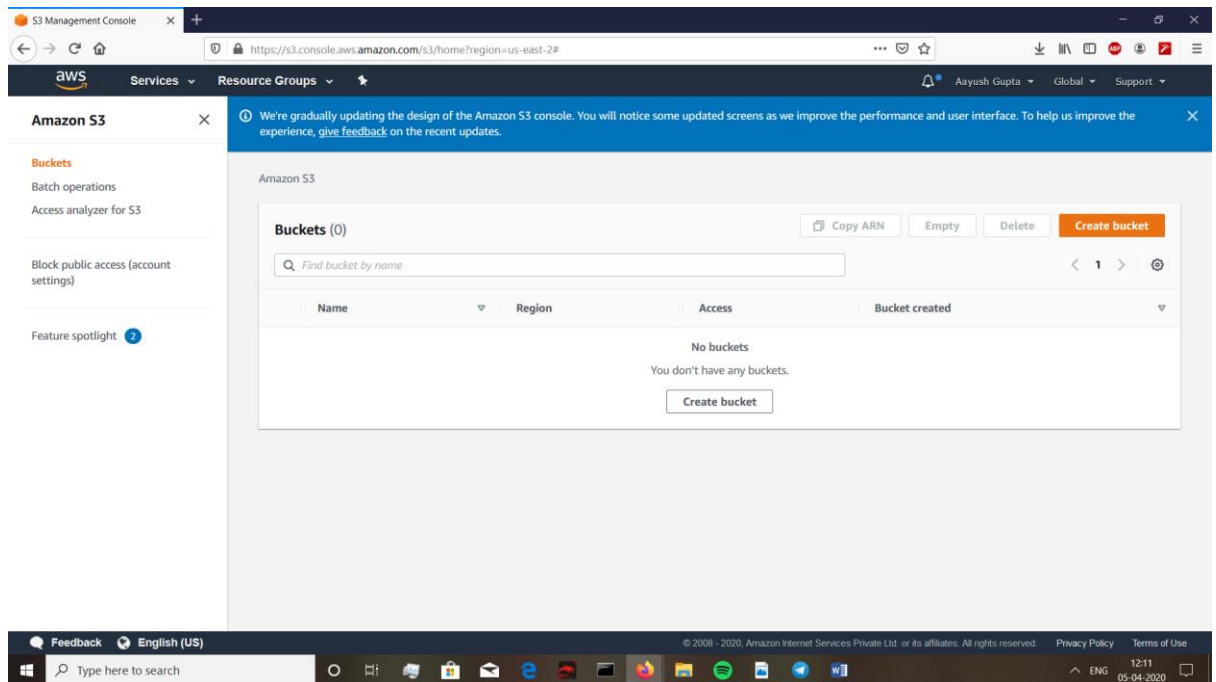
1. AWS Login screen with username



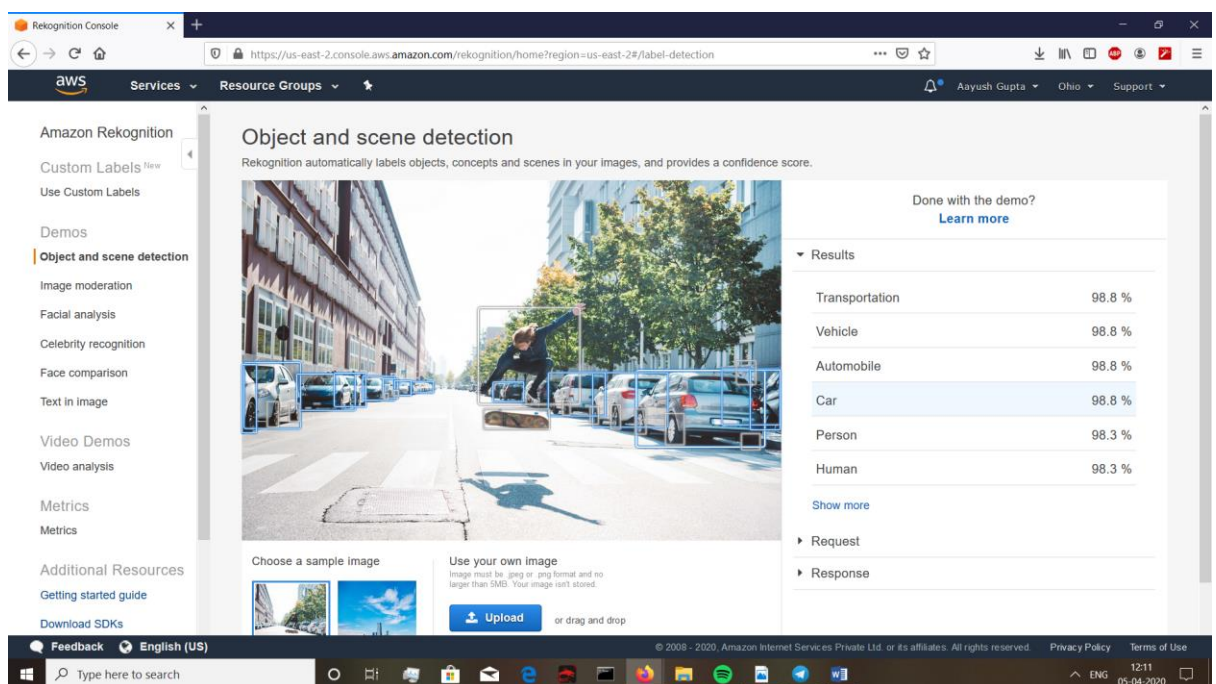
2. EC2 Dashboard



3. S3 Dashboard



4. Rekognition Dashboard



Screenshots needed for EC2

1. Choosing an AMI

Launch instance wizard | EC2 M X +

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 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Search for an AMI by entering a search term e.g. "Windows"

Quick Start

My AMIs

AWS Marketplace

Community AMIs

Free tier only

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0e01ce4ee18447327 (64-bit x86) / ami-03201f374ab66a26e (64-bit Arm)

Amazon Linux 2.29.1, 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

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

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

64-bit (x86)

64-bit (Arm)

Feedback English (US)

Type here to search

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

12:14 05-04-2020

2. Choosing an Instance Type

Launch instance wizard | EC2 M X +

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

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
<input type="checkbox"/>	General purpose	t3a.nano	2	0.5	EBS only	Yes	Up to 5 Gigabit	Yes

Cancel Previous Review and Launch Next: Configure Instance Details

Feedback English (US)

Type here to search

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

12:14 05-04-2020

3. Adding Storage

The screenshot shows the AWS Management Console's EC2 Launch Wizard at Step 4: Add Storage. The breadcrumb trail at the top indicates the steps: 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage (current), 5. Add Tags, 6. Configure Security Group, and 7. Review. The main heading is "Step 4: Add Storage". Below it, a note states: "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

Below the table is a button labeled "Add New Volume". A blue information box contains the text: "Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions."

At the bottom right, there are navigation buttons: "Cancel", "Previous", "Review and Launch" (highlighted), and "Next: Add Tags".

4. Configuring Security Group

The screenshot shows the AWS Management Console's EC2 Launch Wizard at Step 6: Configure Security Group. The breadcrumb trail at the top indicates the steps: 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage, 5. Add Tags, 6. Configure Security Group (current), and 7. Review. The main heading is "Step 6: Configure Security Group". Below it, a note states: "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."

Under "Assign a security group:", there are two radio buttons: "Create a new security group" (selected) and "Select an existing security group".

Below the radio buttons, there are input fields for "Security group name:" (containing "launch-wizard-1") and "Description:" (containing "launch-wizard-1 created 2020-04-05T12:14:53.658+05:30").

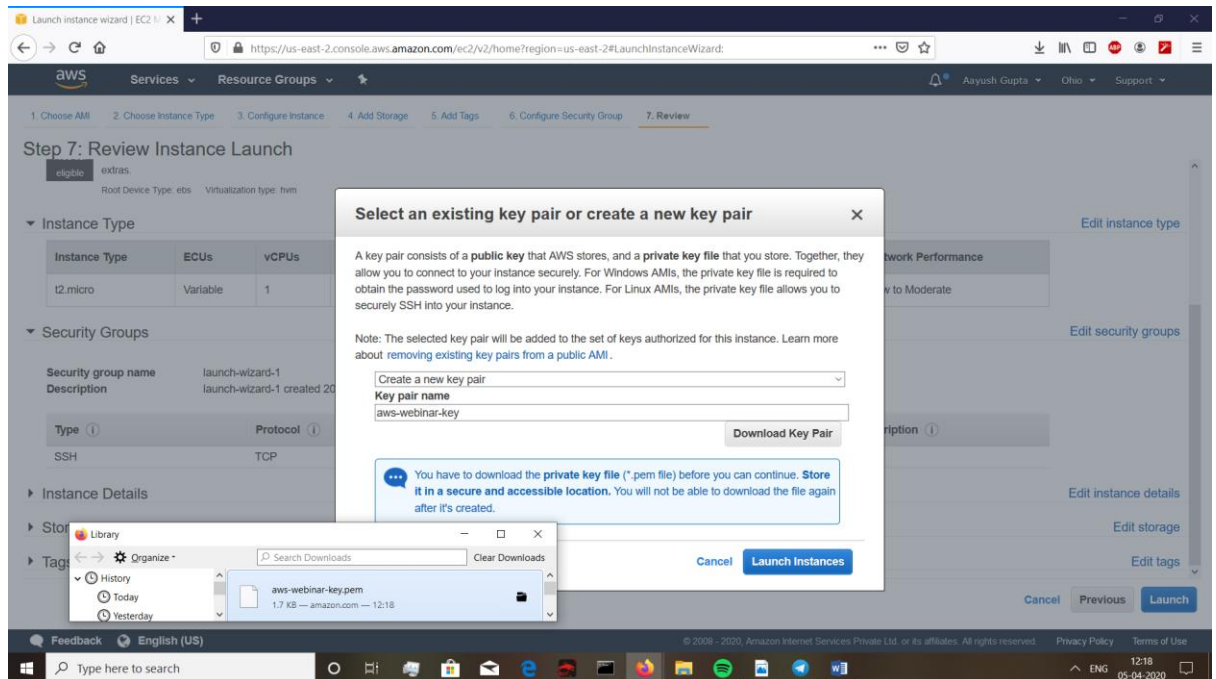
Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop

Below the table is a button labeled "Add Rule".

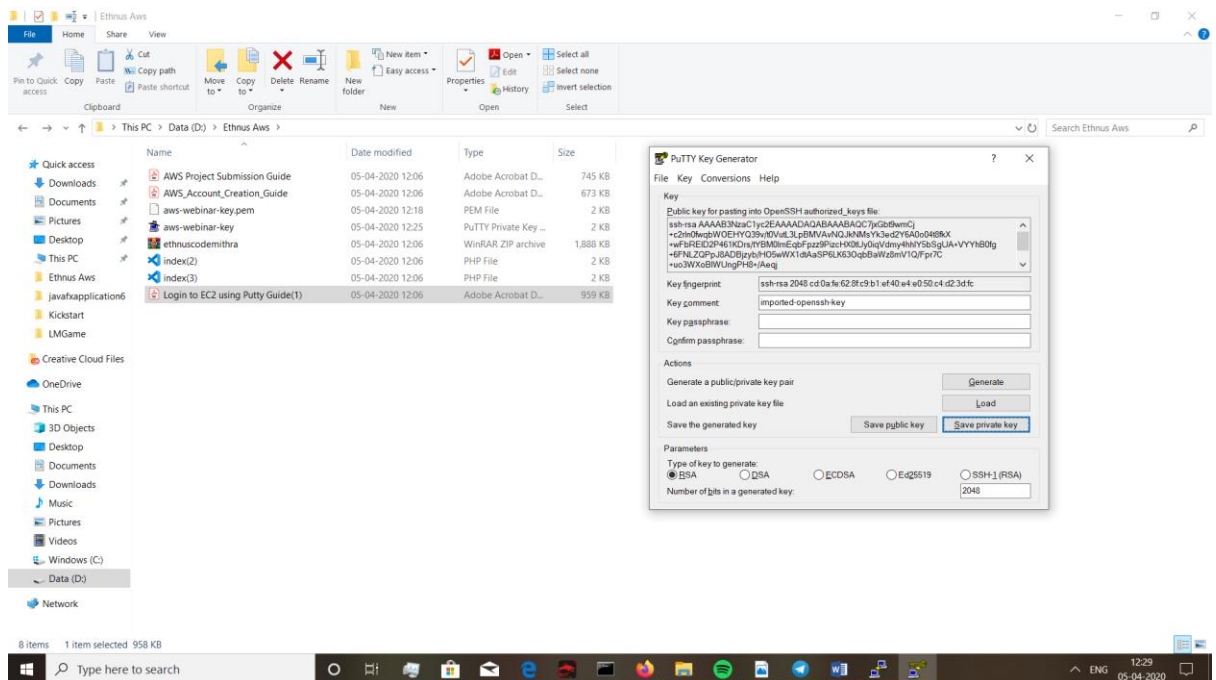
A yellow warning box contains the text: "Warning Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only."

At the bottom right, there are navigation buttons: "Cancel", "Previous", "Review and Launch" (highlighted), and "Next: Add Tags".

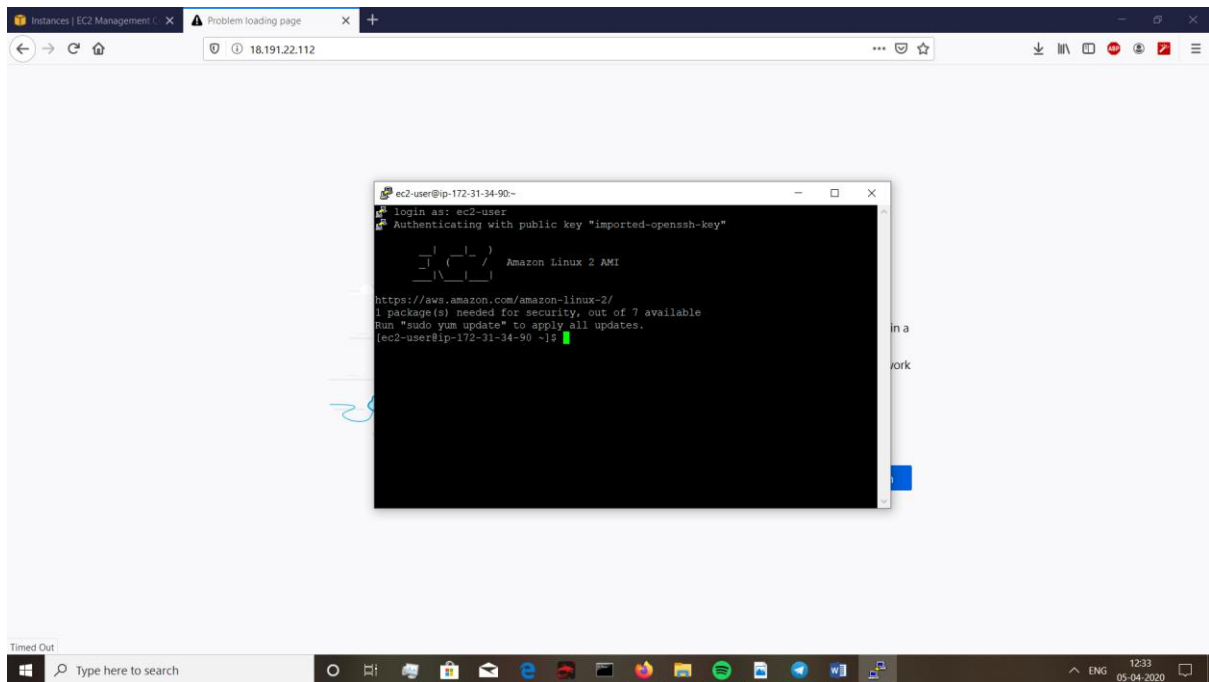
5. Key Pair Download



6. PuTTYgen conversion from pem to ppk

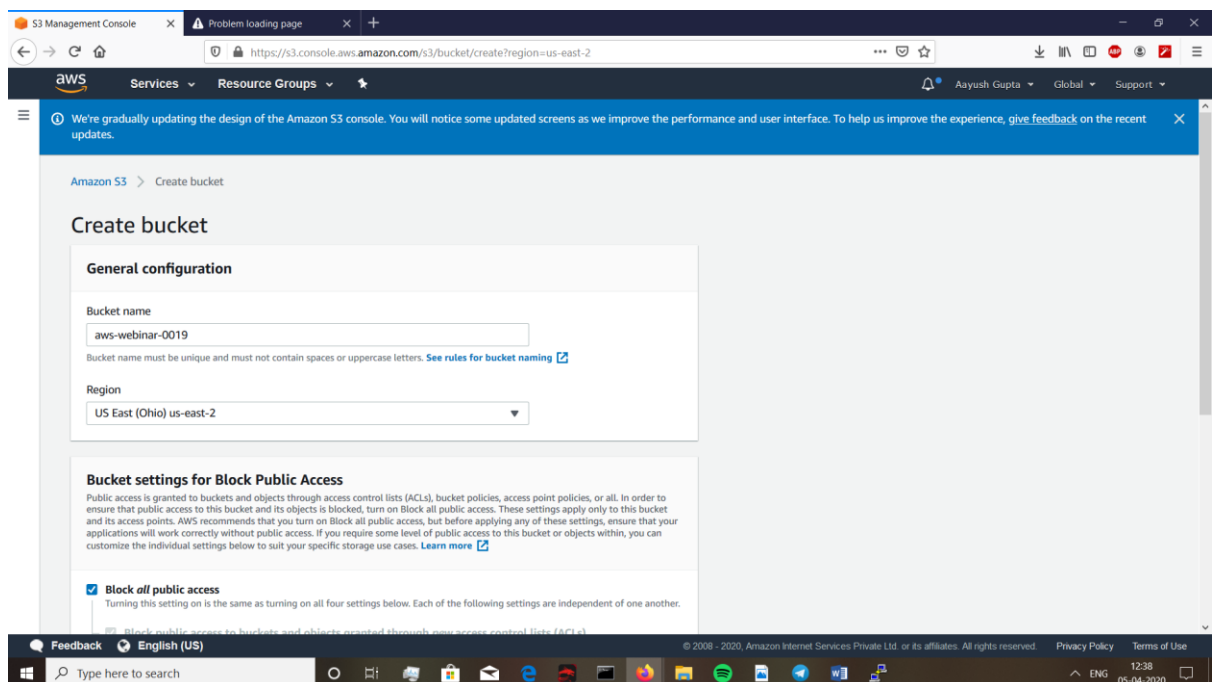


7. Logged in EC2 black screen

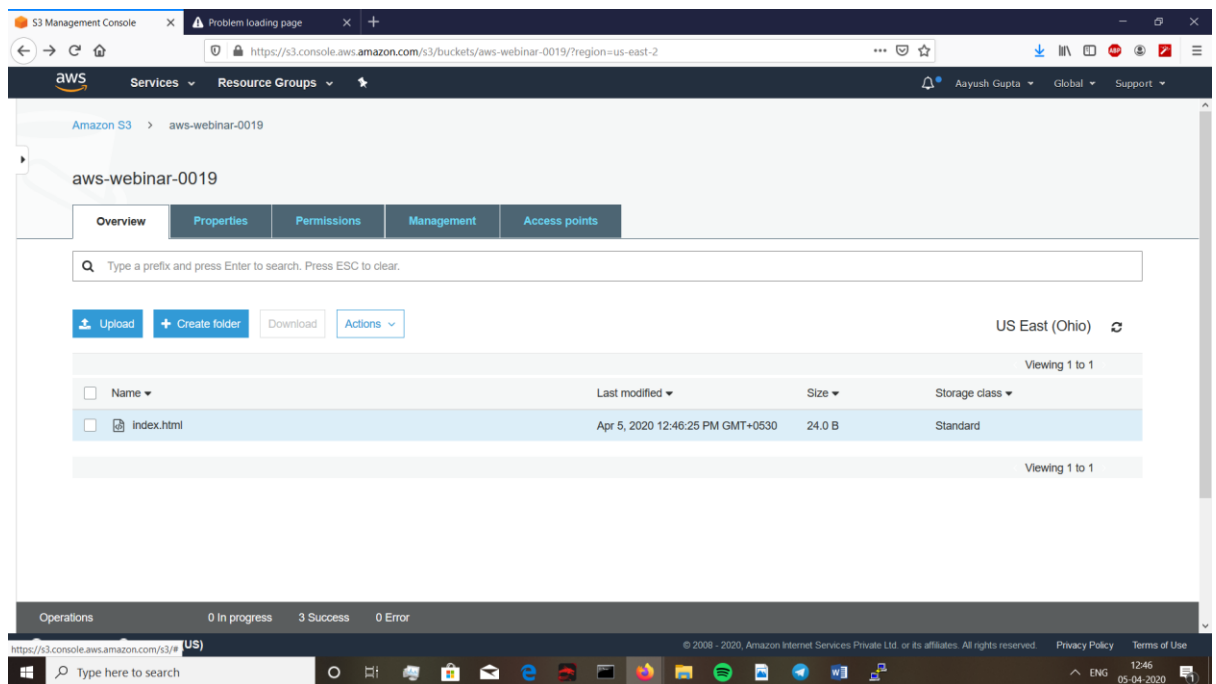


Screenshots needed for S3

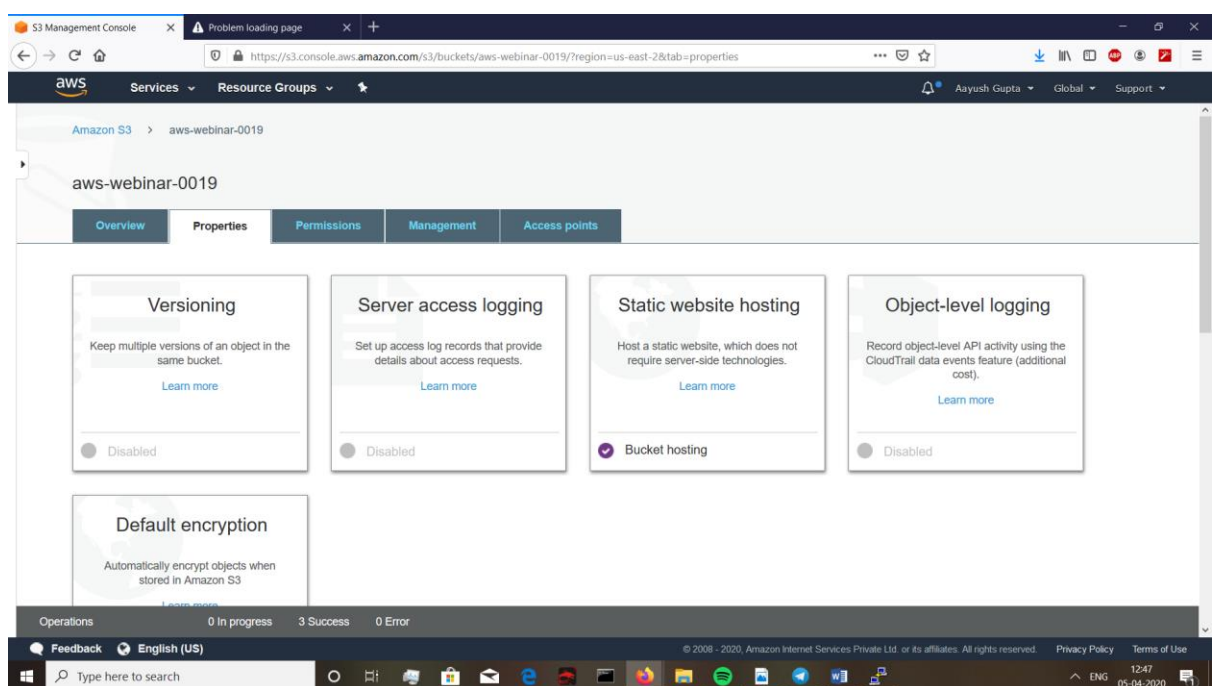
1. Creating a bucket



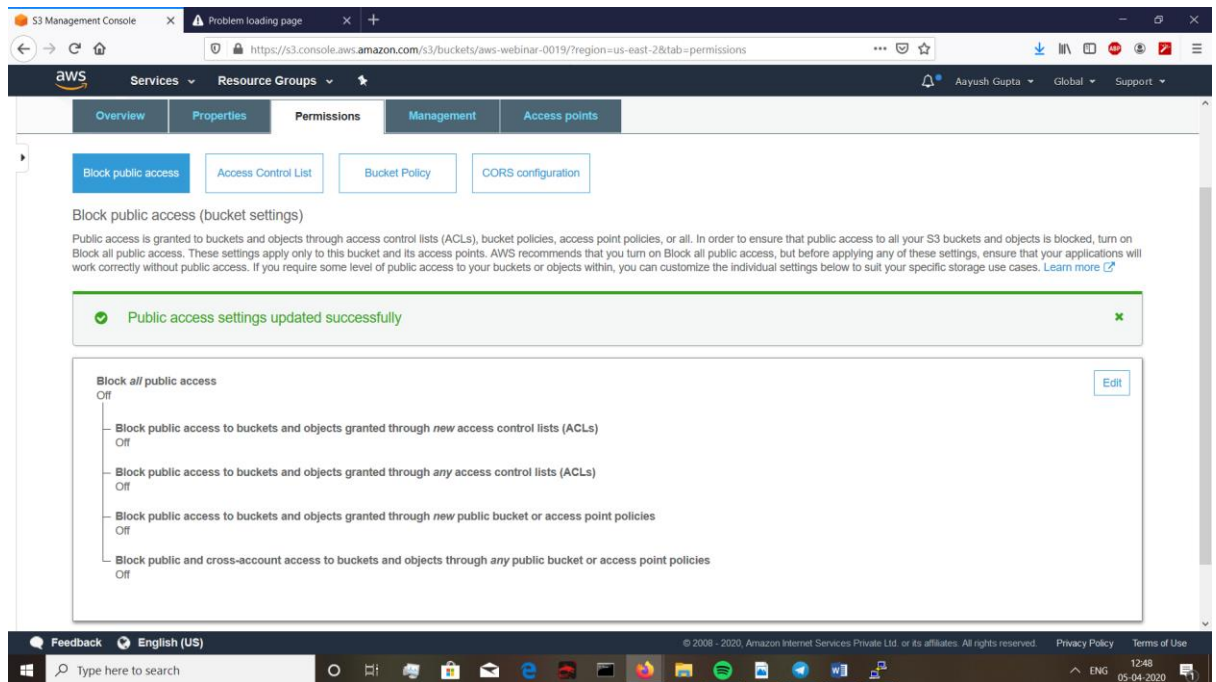
2. Uploading an Object



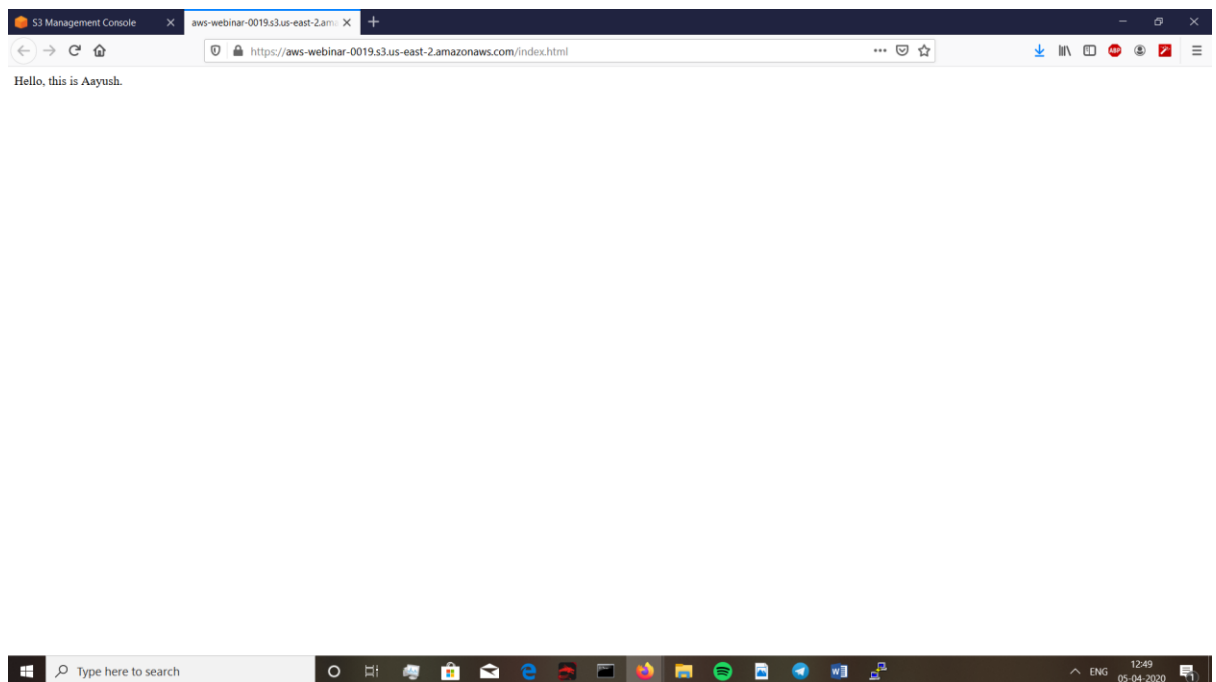
3. Enabling Static Website



4. Making the Object Public

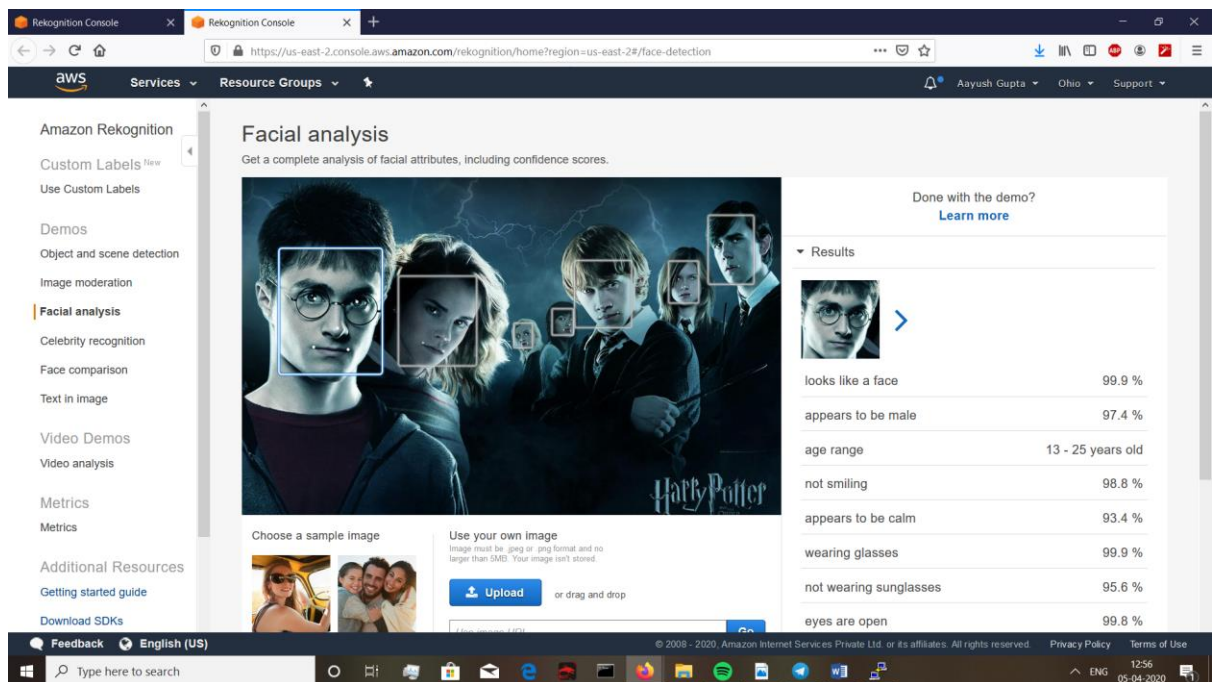


5. Checking the S3 link on the browser

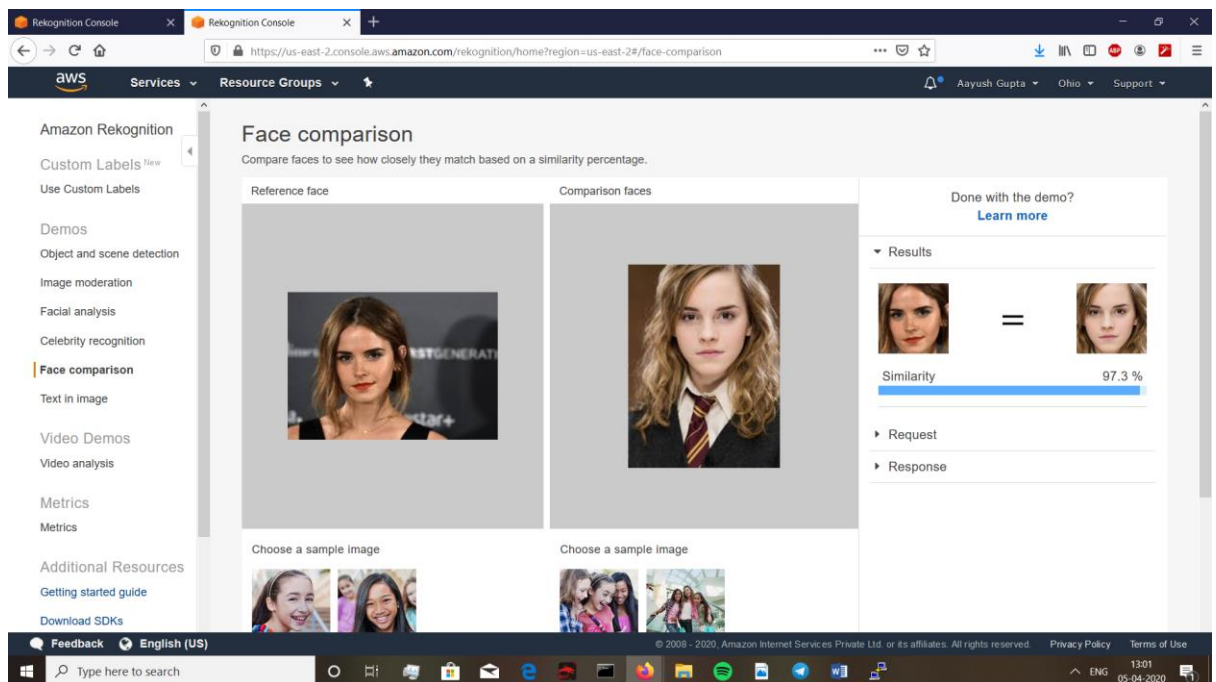


Screenshots needed for Rekognition

1. Face Detect



2. Face Compare



3. Celebrity Recognition

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

Celebrity recognition

Rekognition automatically recognizes celebrities in images and provides confidence scores.

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

▼ Results

 **Emma Watson**
[Learn More](#)

Match confidence 100 %

► Request
► Response

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.

[Upload](#) or drag and drop

Use image URL [Go](#)

Feedback English (US)

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

4. Text in Image

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
Forum

Text in image

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

▼ Results

US English only

WE | de | not | NEED
MAGIC
to | the
CHANGE | WORLD
We CARRY ALL THE
POWER
WE NEED OURSELVES
INSIDE | already
WE HAVE THE
POWER to IMAGINE
BETTER
J.K. ROWLING

► Request
► Response

Feedback English (US)

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

Screenshots needed for EC2 & S3

1. Installing aws-sdk

```
ec2-user@ip-172-31-34-90:/var/www/html/face
# Use 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

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] [--] []...

[ec2-user@ip-172-31-34-90 face]$
[ec2-user@ip-172-31-34-90 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 GiB) copied, 13.4048 s, 80.1 MB/s
[ec2-user@ip-172-31-34-90 face]$ sudo /sbin/swap /var/swap.1
swapon: /var/swap.1: insecure permissions 0644, 0600 suggested.
Setting up swapspace version 1, size = 1024 MiB (1073737728 bytes)
no label, UUID=4095411c-15f5-48c5-8c3d-90259ca71aa5
[ec2-user@ip-172-31-34-90 face]$ sudo /sbin/swapoff /var/swap.1
swapoff: /var/swap.1: insecure permissions 0644, 0600 suggested.
[ec2-user@ip-172-31-34-90 face]$
[ec2-user@ip-172-31-34-90 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 (v3.9.3): 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)
GuzzleHttp/Guzzle is abandoned, you should avoid using it. Use guzzlehttp/guzzle instead.
Writing lock file
Generating autoload files
[ec2-user@ip-172-31-34-90 face]$
```

2. Installing php

```
ec2-user@ip-172-31-34-90:/var/www/html/face
Total 23 MB/s | 6.6 MB 00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Installing : apr-1.6.3-5.amzn2.0.2.x86_64 1/13
Installing : apr-util-bdb-1.6.1-5.amzn2.0.2.x86_64 2/13
Installing : apr-util-1.6.1-5.amzn2.0.2.x86_64 3/13
Installing : httpd-tools-2.4.41-1.amzn2.0.1.x86_64 4/13
Installing : httpd filesystem-2.4.41-1.amzn2.0.1.noarch 5/13
Installing : mailcap-2.1.41-2.amzn2.noarch 6/13
Installing : generic-logos-httpd-19.0.0-4.amzn2.noarch 7/13
Installing : mod http2-1.15.3-2.amzn2.x86_64 8/13
Installing : httpd-2.4.41-1.amzn2.0.1.x86_64 9/13
Installing : libzip010-compat-0.10.1-9.amzn2.0.5.x86_64 10/13
Installing : php-common-5.4.16-46.amzn2.0.2.x86_64 11/13
Installing : php-5.4.16-46.amzn2.0.2.x86_64 12/13
Installing : php-5.4.16-46.amzn2.0.2.x86_64 13/13
Verifying : apr-util-1.6.1-5.amzn2.0.2.x86_64 1/13
Verifying : libzip010-compat-0.10.1-9.amzn2.0.5.x86_64 2/13
Verifying : apr-util-bdb-1.6.1-5.amzn2.0.2.x86_64 3/13
Verifying : php-5.4.16-46.amzn2.0.2.x86_64 4/13
Verifying : mod http2-1.15.3-2.amzn2.x86_64 5/13
Verifying : generic-logos-httpd-19.0.0-4.amzn2.noarch 6/13
Verifying : httpd-2.4.41-1.amzn2.0.1.x86_64 7/13
Verifying : php-5.4.16-46.amzn2.0.2.x86_64 8/13
Verifying : apr-1.6.3-5.amzn2.0.2.x86_64 9/13
Verifying : mailcap-2.1.41-2.amzn2.noarch 10/13
Verifying : httpd filesystem-2.4.41-1.amzn2.0.1.noarch 11/13
Verifying : httpd-tools-2.4.41-1.amzn2.0.1.x86_64 12/13
Verifying : php-common-5.4.16-46.amzn2.0.2.x86_64 13/13

Installed:
php.x86_64 0:5.4.16-46.amzn2.0.2

dependency Installed:
apr.x86_64 0:1.6.3-5.amzn2.0.2
apr-util.x86_64 0:1.6.1-5.amzn2.0.2
apr-util-bdb.x86_64 0:1.6.1-5.amzn2.0.2
generic-logos-httpd.noarch 0:19.0.0-4.amzn2
httpd.x86_64 0:2.4.41-1.amzn2.0.1
httpd filesystem.noarch 0:2.4.41-1.amzn2.0.1
httpd-tools.x86_64 0:2.4.41-1.amzn2.0.1
libzip010-compat.x86_64 0:0.10.1-9.amzn2.0.5
mailcap.noarch 0:2.1.41-2.amzn2
mod http2.x86_64 0:1.15.3-2.amzn2
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!
```

3. index.php file code

```
ec2-user@ip-172-31-34-90:/var/www/html/face
$ ./face/face.php
$ sudo mv face
$ cd face
$ sudo php -d memory_limit=-1 ~/composer.phar require aws/aws-sdk-php
$
$ ls 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/b97ea33b5842c7894b8049236c05580.jpg
$ sudo mv b97ea33b5842c7894b8049236c05580.jpg sample.jpg
$
$ error_reporting(0);
$ require_once( __DIR__ . '/vendor/autoload.php');
$ use Aws\S3\S3Client;
$ use Aws\Rekognition\RekognitionClient;
$
$ $bucket = 'aws-webinar-0019';
$ $keyname = 'hp.jpg';
$
$ $s3 = S3Client::factory([
$     'profile' => 'default',
$     'region' => 'us-east-1',
$     '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;
$ }
$
```

5. Upload success screenshot

```
ec2-user@ip-172-31-34-90:/var/www/html/face
$ no label, UUID=4095411c-15f5-48c5-8c3d-90259ca71aa5
$ [ec2-user@ip-172-31-34-90 face]$ sudo /sbin/swapon /var/swap.1
$ swapon: /var/swap.1: insecure permissions 0644, 0600 suggested.
$ [ec2-user@ip-172-31-34-90 face]$
$ [ec2-user@ip-172-31-34-90 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 (v3.9.3): 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 3 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)
$ Guzzle/guzzle is abandoned, you should avoid using it. Use guzzlehttp/guzzle instead.
$ Writing lock file
$ Generating autoload files
$ [ec2-user@ip-172-31-34-90 face]$ sudo wget https://www.elsegete.cat/myimg/t/92-925016_harry-potter-and-the-deathly-hallows-4k.jpg
$ --2020-04-05 07:50:44-- https://www.elsegete.cat/myimg/t/92-925016_harry-potter-and-the-deathly-hallows-4k.jpg
$ Resolving www.elsegete.cat (www.elsegete.cat)... 204.12.226.182
$ Connecting to www.elsegete.cat (www.elsegete.cat)|204.12.226.182|:443... connected.
$ HTTP request sent, awaiting response... 200 OK
$ Length: 201256 (197K) [image/jpeg]
$ Saving to: '92-925016_harry-potter-and-the-deathly-hallows-4k.jpg'
$
$ 100%[=====] 201,256  --K/s  in 0.06s
$
$ 2020-04-05 07:50:44 (3.31 MB/s) - '92-925016_harry-potter-and-the-deathly-hallows-4k.jpg' saved [201256/201256]
$
$ [ec2-user@ip-172-31-34-90 face]$ sudo mv ^C
$ [ec2-user@ip-172-31-34-90 face]$ ^C
$ [ec2-user@ip-172-31-34-90 face]$ sudo mv 92-925016_harry-potter-and-the-deathly-hallows-4k.jpg hp.jpg
$ [ec2-user@ip-172-31-34-90 face]$ ls
$ composer.json  composer.lock  hp.jpg  vendor
$ [ec2-user@ip-172-31-34-90 face]$ sudo vim index.php
$ [ec2-user@ip-172-31-34-90 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-172-31-34-90 face]$ sudo php index.php
$ Image upload done... Here is the URL: https://aws-webinar-0019.s3.us-east-2.amazonaws.com/hp.jpg[ec2-user@ip-172-31-34-90 face]$
```

ETHNUS AWS MASTERCLASS

The screenshot displays the AWS S3 Management Console interface. The main view shows the 'aws-webinar-0019' bucket with a list of objects. The 'hp.jpg' file is selected, and a details panel is open on the right.

Bucket: aws-webinar-0019

Overview | Properties | Permissions | Management | Access points

Search: Type a prefix and press Enter to search. Press ESC to clear.

Buttons: Upload, Create folder, Download, Actions

Name	Last modified
hp.jpg	Apr 5, 2020 1:38:56 PM GMT+0530
index.html	Apr 5, 2020 12:46:25 PM GMT+0530

hp.jpg details panel:

- Download | Copy path | Select from
- Latest version
- Overview**
 - Key: hp.jpg
 - Size: 196.5 KB
 - Expiration date: N/A
 - Expiration rule: N/A
 - ETag: 5c4658d8d3860e35774922dc8414424fe
 - Last modified: Apr 5, 2020 1:38:56 PM GMT+0530
 - Object URL: <https://aws-webinar-0019.s3.us-east-2.amazonaws.com/hp.jpg>
- Properties**
 - Storage class: Standard
 - Encryption: None
 - Metadata: 1
 - Tags: 0 Tags
 - Object lock: Disabled
- Permissions**
 - Owner

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

13:39 05-04-2020