

<Codemithra />™



Explore | Expand | Enrich

Building Face-detection App on AWS

Under guidance of:

Aravind Raghavan

AWS Certified Cloud Expert

Presented by: **Lathesh Neikar K H**

Usn:4SU16CS051

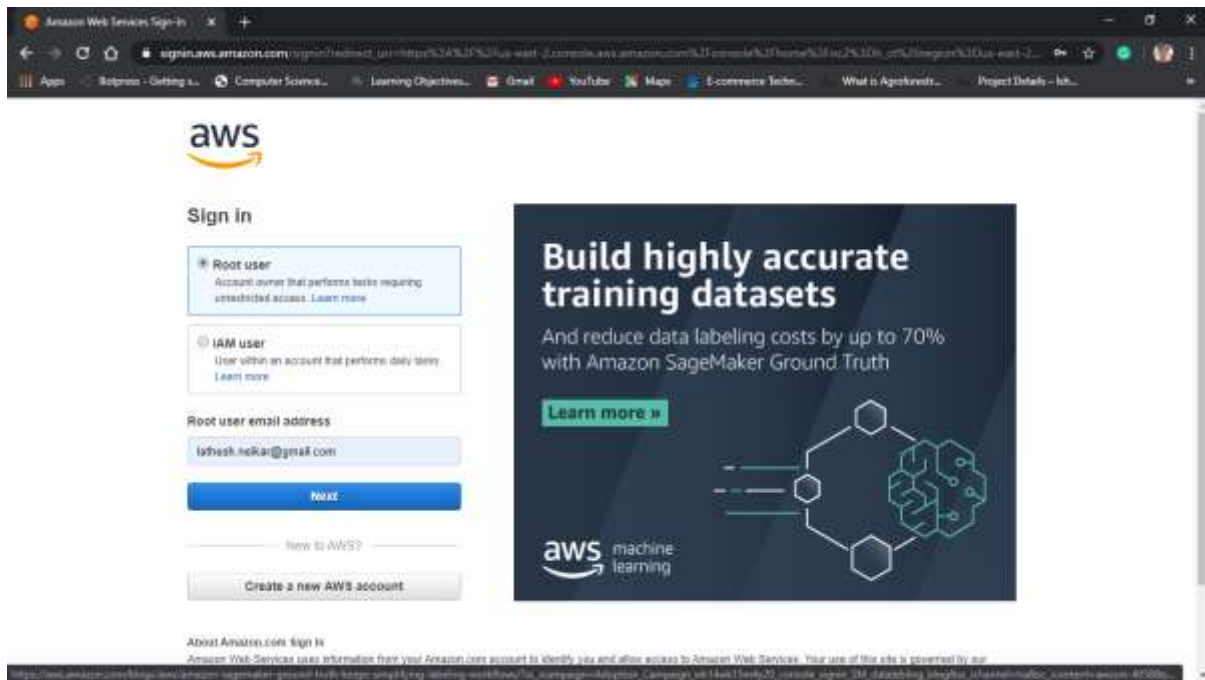
Phone:9380932584

Email : lathesh.neikar@gmail.com

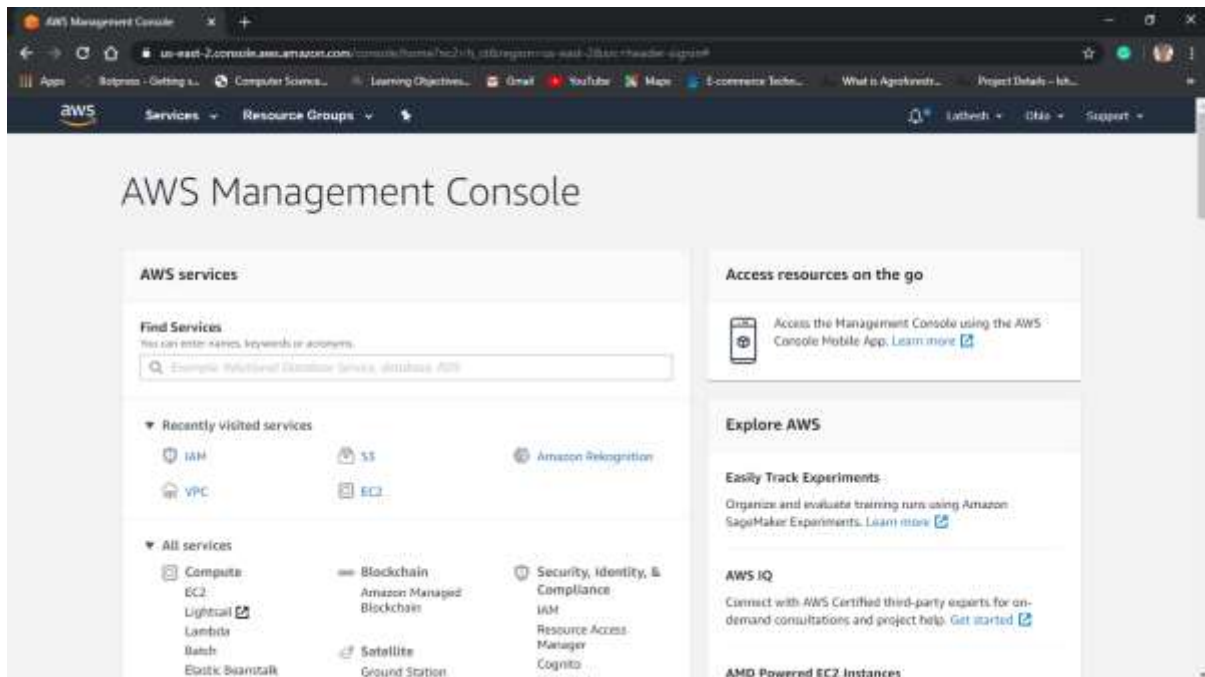
College: SDM institute of technology, Ujire DK

Screenshots

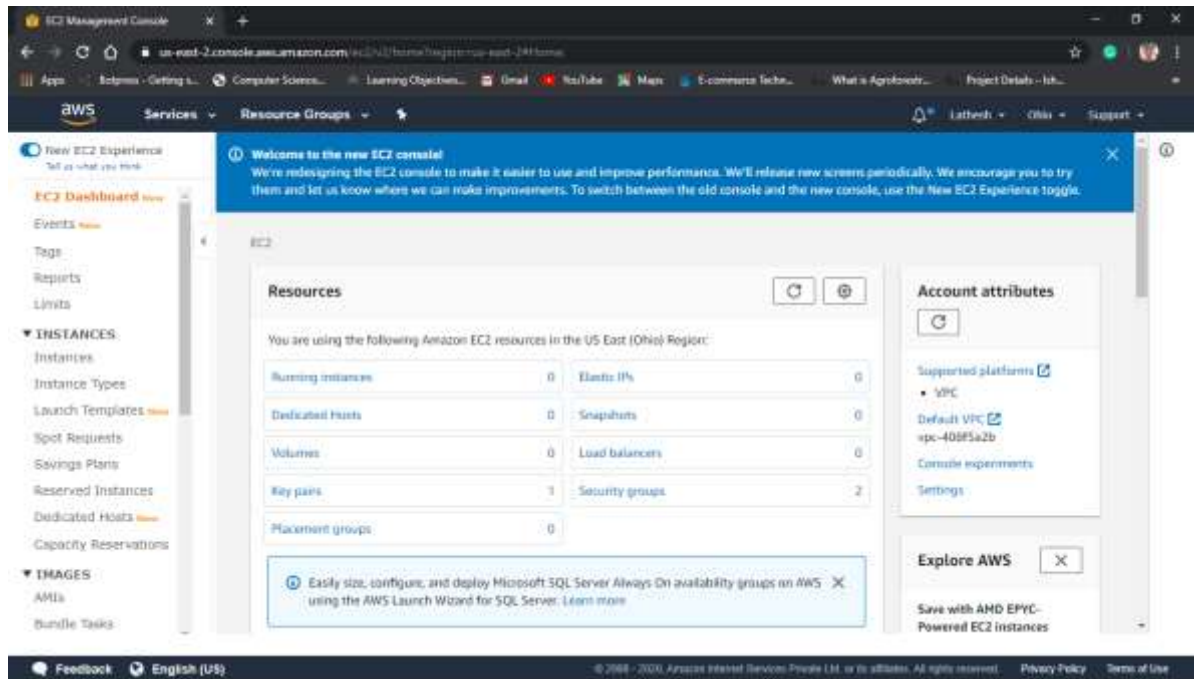
1. AWS login screen with username



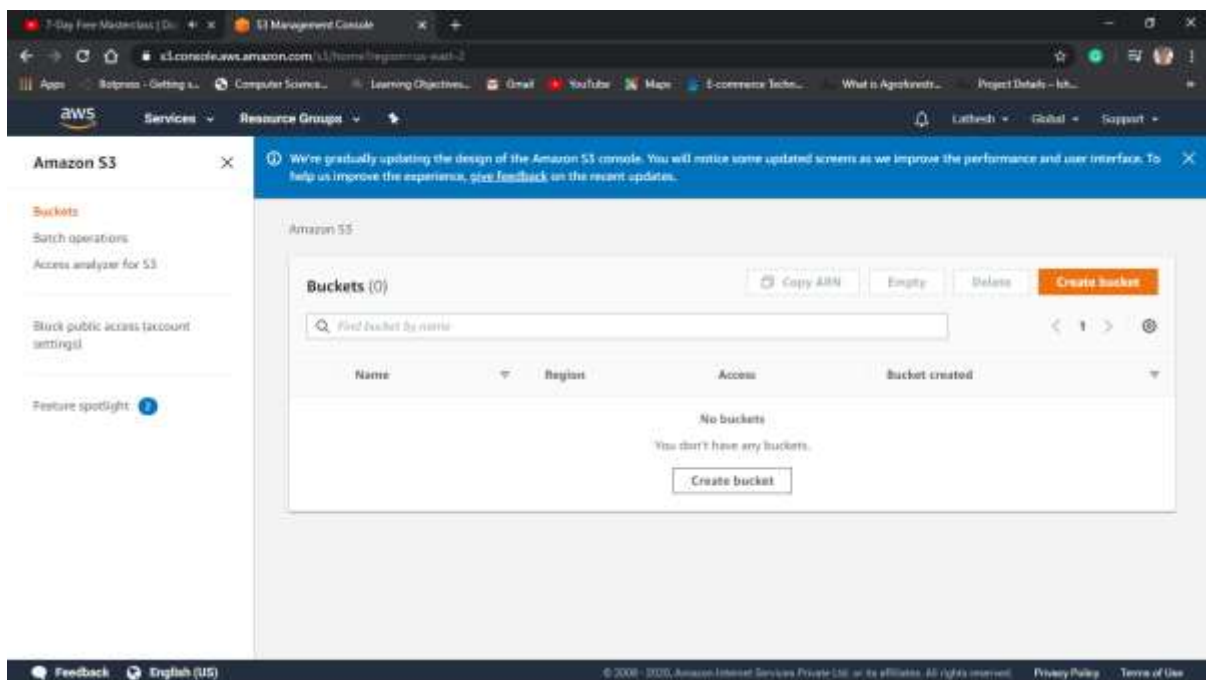
AWS Management Console



2. EC2 Dashboard



3. S3 Dashboard



4.Rekognition Dashboard

The screenshot shows the Amazon Rekognition console in a web browser. The browser's address bar displays the URL `us-east-2.console.aws.amazon.com/rekognition/home?region=us-east-2#`. The console interface includes a top navigation bar with the AWS logo, 'Services', and 'Resource Groups'. A left-hand sidebar lists various features: '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', and 'Metrics'. The main content area features a large header with the title 'Amazon Rekognition' and the description 'Deep learning-based visual analysis service. Search, verify, and organize millions of images and videos.' Below this header are two buttons: 'Try Demo' and 'Download SDKs'. The lower section of the dashboard is divided into three columns, each with an icon and a heading: 1) 'Easily Integrate Powerful Visual Analysis into Your App' with a stack of layers icon, stating 'You don't need computer vision or deep learning expertise to take advantage of'; 2) 'Continuously Learning' with a circuit icon, stating 'Amazon Rekognition is designed to use deep learning technology to analyze billions of images and videos daily. It is'; 3) 'Integrated with AWS Services' with a puzzle pieces icon, stating 'Amazon Rekognition is designed to work seamlessly with other AWS services. Rekognition integrates directly with Amazon'. The footer contains a 'Feedback' link, 'English (US)' language selector, and copyright information: '© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.' along with links to 'Privacy Policy' and 'Terms of Use'.

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

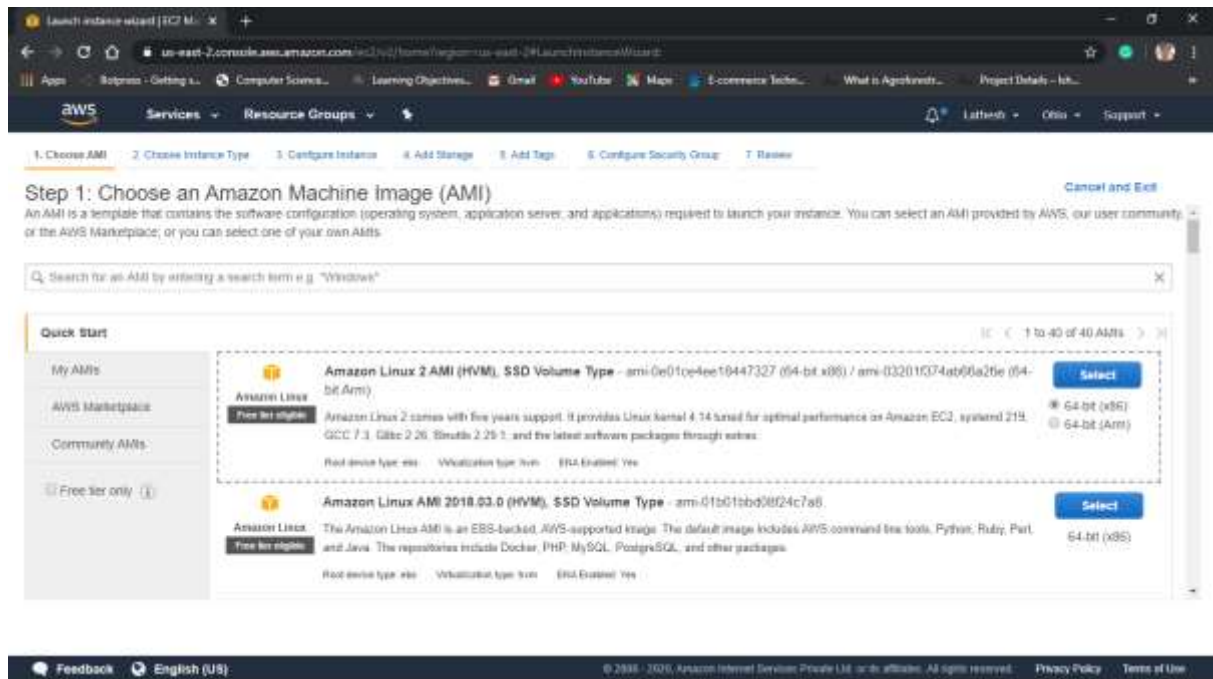
Continuously Learning
Amazon Rekognition is designed to use deep learning technology to analyze billions of images and videos daily. It is

Integrated with AWS Services
Amazon Rekognition is designed to work seamlessly with other AWS services. Rekognition integrates directly with Amazon

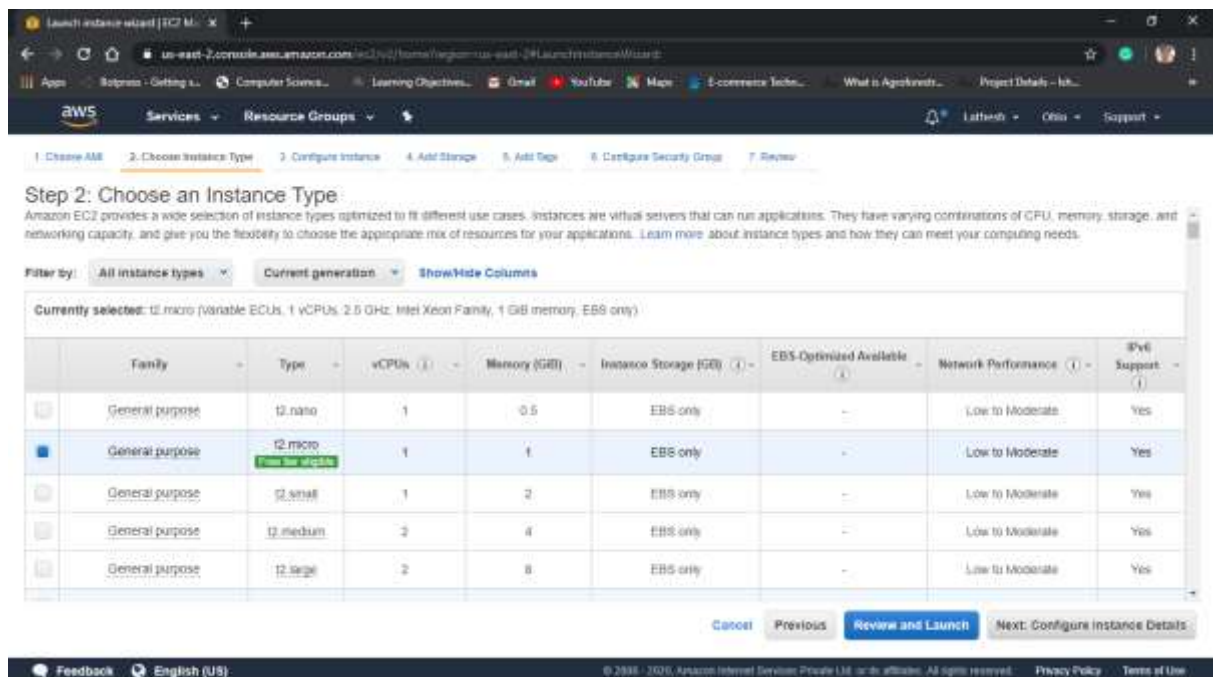
Feedback English (US) © 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

SCREENSHOTS FOR EC2

1. Choosing an AMI



2. Choosing an Instance type



3. Adding Storage

The screenshot shows the 'Step 4: Add Storage' page of the AWS Launch Instance Wizard. The wizard progress bar at the top indicates the current step is '4. Add Storage'. The page title is 'Step 4: Add Storage'. Below the title, a paragraph explains that the instance will be launched with the following storage device settings and that additional EBS volumes can be attached after launch. A table lists the storage configuration for the root volume. The table has columns: Volume Type, Device, Snapshot, Size (GiB), Volume Type, IOPS, Throughput (MB/s), Delete on Termination, and Encryption. The root volume is configured with a device of /dev/xvda, snapshot snap-0f54632056aas4c20, a size of 8 GiB, a volume type of General Purpose SSD (gp2), IOPS of 100 / 5000, throughput of N/A, and encryption is not enabled. Below the table is an 'Add New Volume' button. A blue box contains a note about free tier eligibility. At the bottom, there are 'Cancel', 'Previous', 'Review and Launch', and 'Next: Add Tags' buttons.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/xvda	snap-0f54632056aas4c20	8	General Purpose SSD (gp2)	100 / 5000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

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

The screenshot shows the 'Step 6: Configure Security Group' page of the AWS Launch Instance Wizard. The wizard progress bar at the top indicates the current step is '6. Configure Security Group'. The page title is 'Step 6: Configure Security Group'. Below the title, a paragraph explains that a security group is a set of firewall rules that control traffic to and from the instance. A section titled 'Assign a security group:' has two radio buttons: 'Create a new security group' (selected) and 'Select an existing security group'. Below this, there are input fields for 'Security group name' (launch-wizard-2) and 'Description' (launch-wizard-2 created 2020-04-05T14:02:50.463+08:30). A table lists the security group rules. The table has columns: Type, Protocol, Port Range, Source, and Description. One rule is shown with Type SSH, Protocol TCP, Port Range 22, Source Custom 0.0.0.0/0, and Description e.g. SSH for Admin Desktop. Below the table is an 'Add Rule' button. A yellow warning box contains a warning icon and text about allowing access from all IP addresses. At the bottom, there are 'Cancel', 'Previous', 'Review and Launch', and 'Next: Add Tags' buttons.

Assign a security group: ☒ Create a new security group ☐ Select an existing security group

Security group name:

Description:

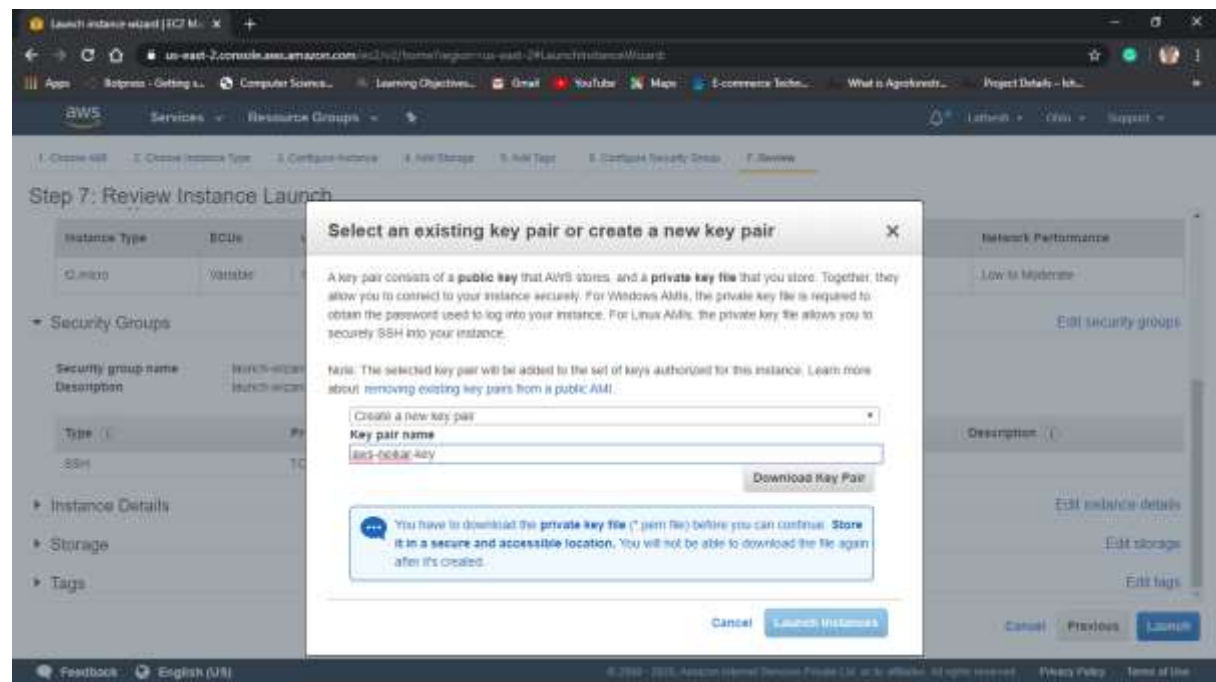
Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop

[Add Rule](#)

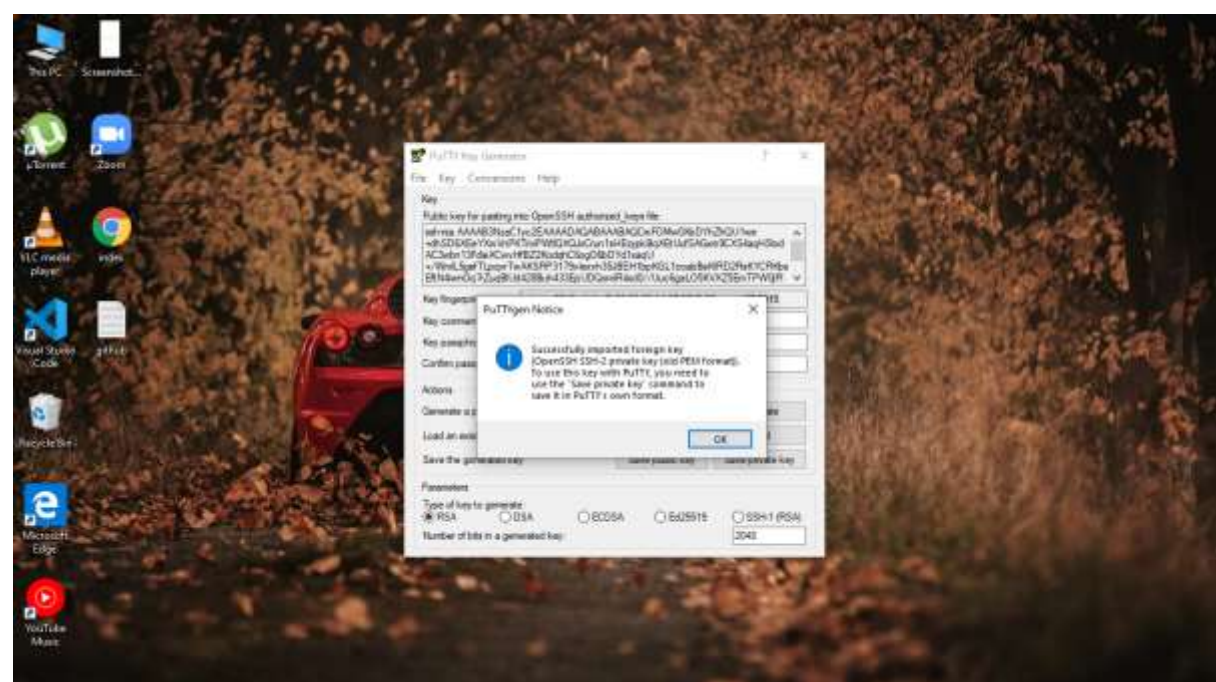
Warning
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Tags](#)

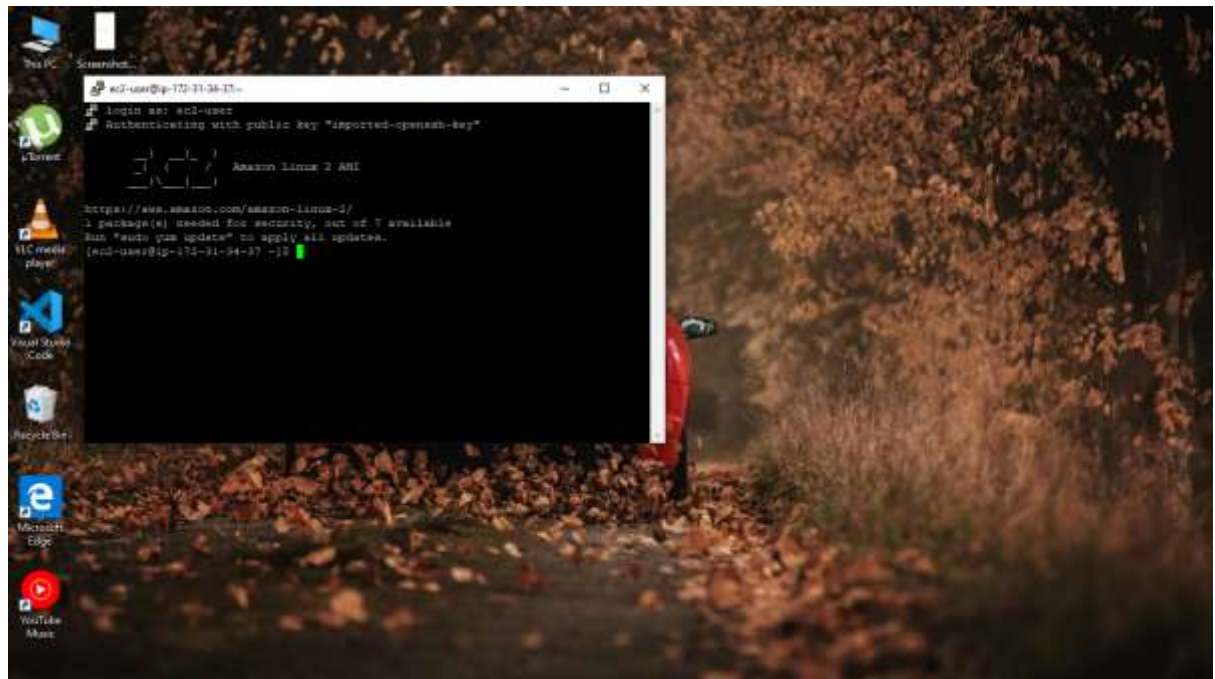
5. Key pair Download



6. PuTTYgen conversion from pem to ppk

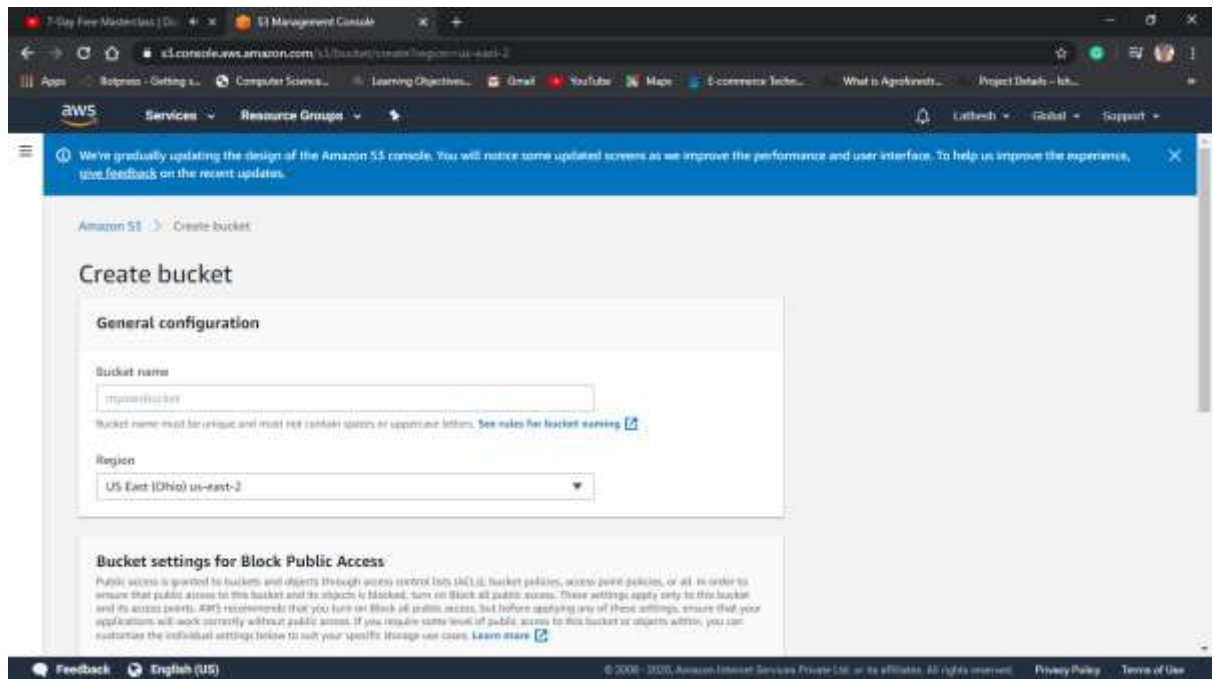


7. Logged in EC2 black screen

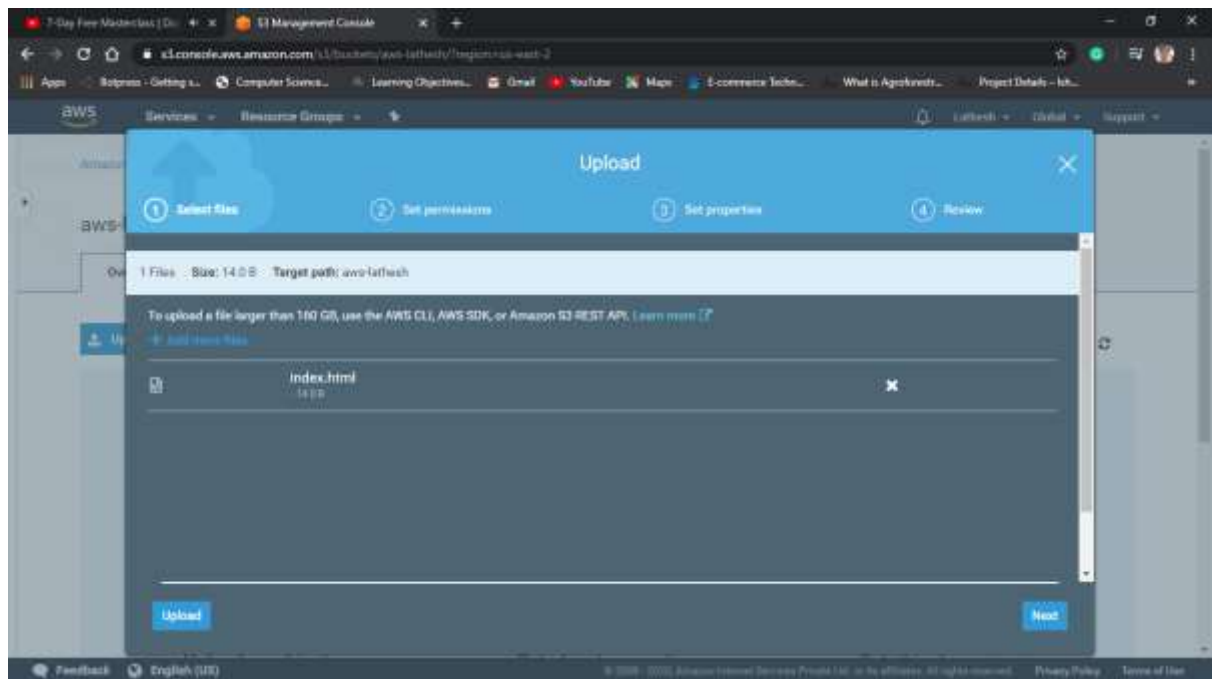


SCREENSHOTS FOR S3

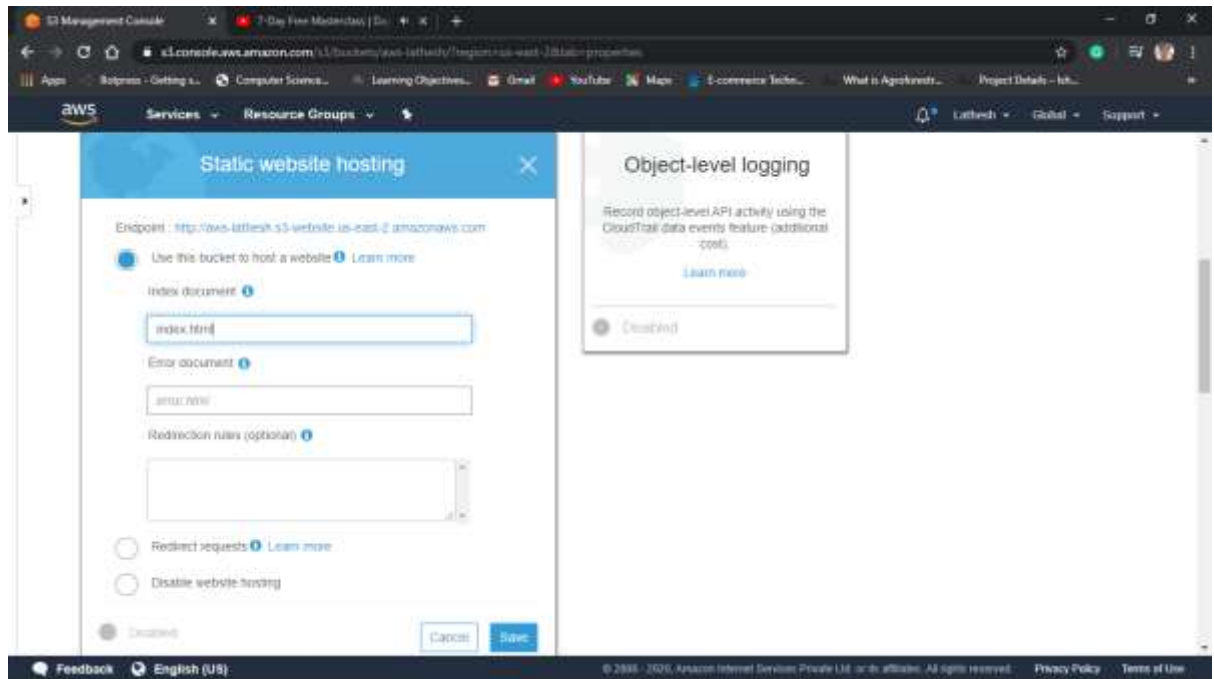
1. Creating a bucket



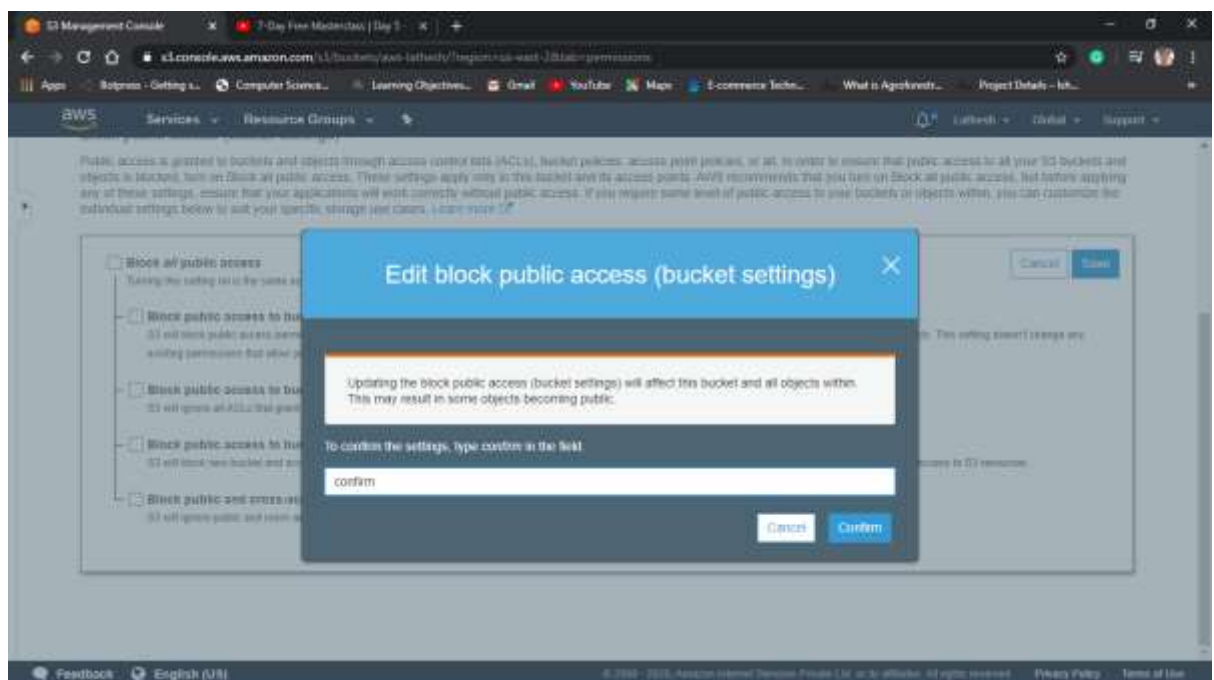
2. Uploading an object



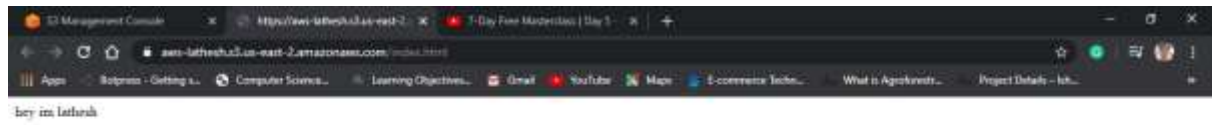
3. Enabling static website



4. Making the object Public

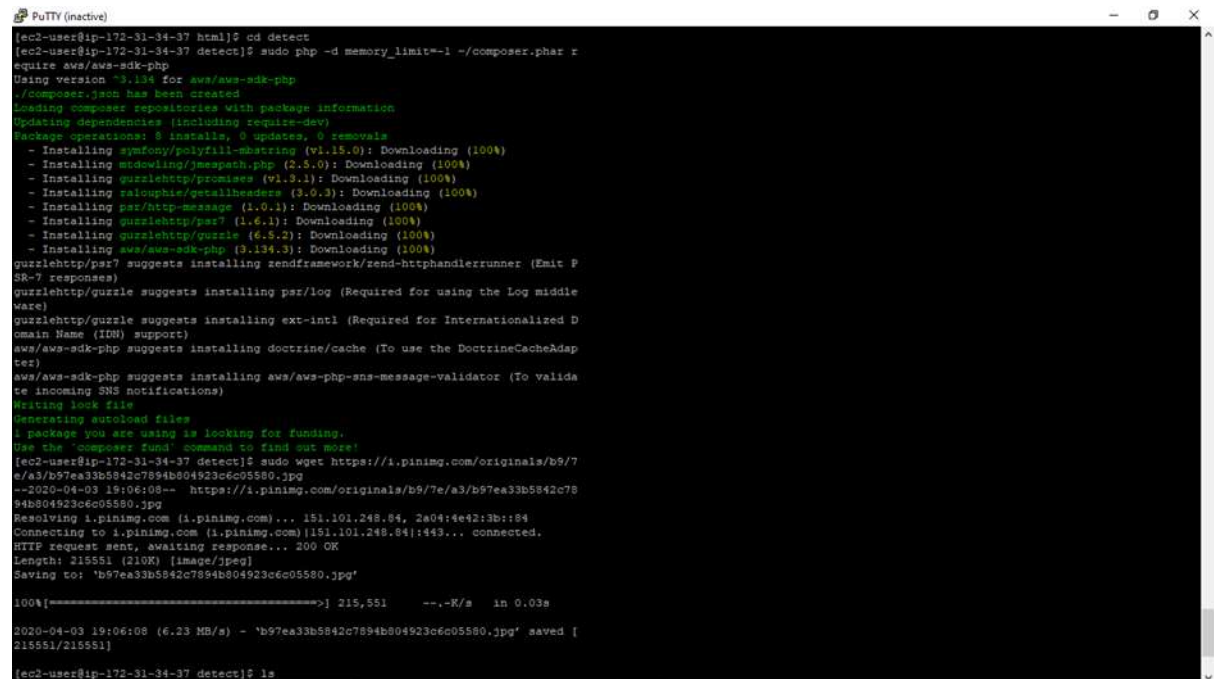


5. Checking S3 link on the browser



SCREENSHOTS FOR EC2 and S3

1.Installing aws-sdk

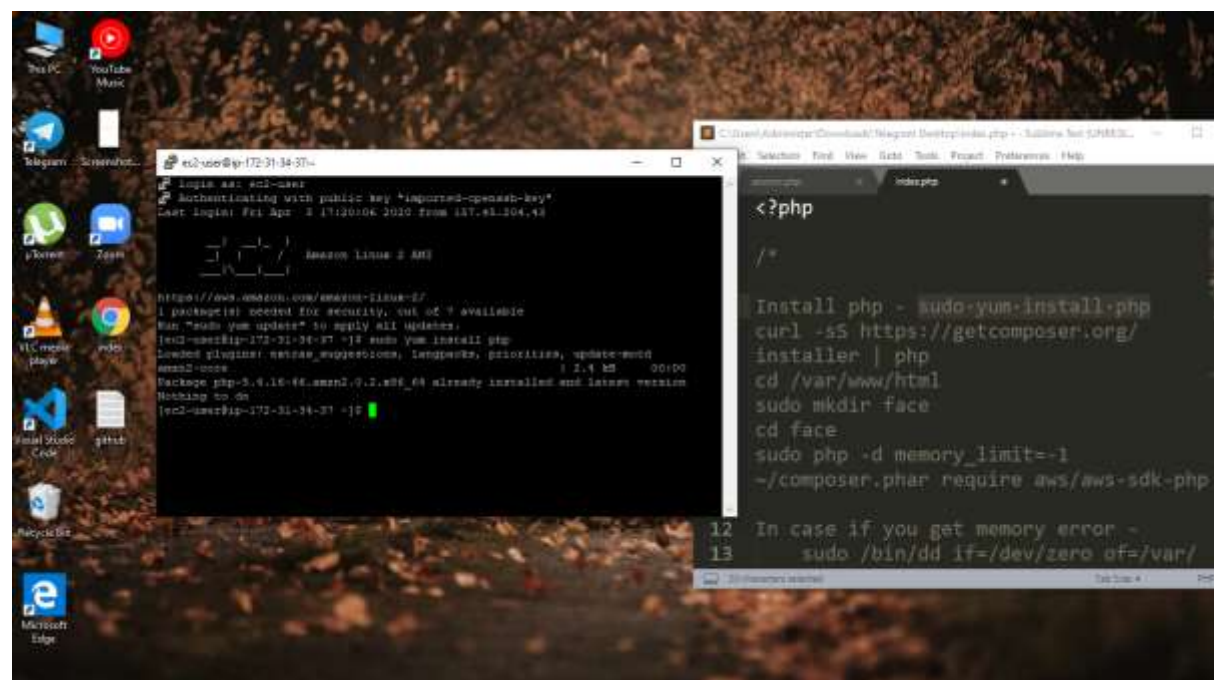


```
[ec2-user@ip-172-31-34-37 html]$ cd detect
[ec2-user@ip-172-31-34-37 detect]$ sudo php -d memory_limit=-1 ~/composer.phar r
require aws/aws-sdk-php
Using version ^3.134 for aws/aws-sdk-php
./composer.json has been created
Loading composer repositories with package information
Updating dependencies (including require-dev)
Package operations: 8 installs, 0 updates, 0 removals
- Installing symfony/polyfill-mbstring (v1.15.0): Downloading (100%)
- Installing zendframework/zend-http (2.5.0): Downloading (100%)
- Installing guzzlehttp/promises (v1.3.1): Downloading (100%)
- Installing ralouphie/getallheaders (3.0.3): Downloading (100%)
- Installing psr/http-message (1.0.1): Downloading (100%)
- Installing guzzlehttp/psr7 (1.6.1): Downloading (100%)
- Installing guzzlehttp/guzzle (6.5.2): Downloading (100%)
- Installing aws/aws-sdk-php (3.134.3): Downloading (100%)
guzzlehttp/psr7 suggests installing zendframework/zend-httphandler (Emit P
SR-7 responses)
guzzlehttp/guzzle suggests installing psr/log (Required for using the Log middle
ware)
guzzlehttp/guzzle suggests installing ext-intl (Required for Internationalized D
omain Name (IDN) support)
aws/aws-sdk-php suggests installing doctrine/cache (To use the DoctrineCacheAdap
ter)
aws/aws-sdk-php suggests installing aws/aws-php-sns-message-validator (To valida
te incoming SNS notifications)
Writing lock file
Generating autoload files
1 package you are using is looking for funding.
Use the "composer fund" command to find out more!
[ec2-user@ip-172-31-34-37 detect]$ sudo wget https://i.pinimg.com/originals/b9/7
e/a3/b97ea33b5842c7894b804923c6c05580.jpg
--2020-04-03 19:06:08-- https://i.pinimg.com/originals/b9/7e/a3/b97ea33b5842c78
94b804923c6c05580.jpg
Resolving i.pinimg.com (i.pinimg.com)... 151.101.248.84, 2a04:4e42:3b::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 0.03s

2020-04-03 19:06:08 (6.23 MB/s) - 'b97ea33b5842c7894b804923c6c05580.jpg' saved [
215551/215551]
[ec2-user@ip-172-31-34-37 detect]$ ls
```

2.Installing php



3.Index.php code

```

root@ec2-user@ip-172-31-34-15:/var/www/html/#detect
# login as: ec2-user
# Authenticating with public key "imported-openssh-key"
Last login: Fri Apr  8 16:00:10 2022 from 107.95.104.14

 _ _ | _ _ | _ _ |
|_| |_| |_| |_| |_| Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/

[ec2-user@ip-172-31-34-27 ~]$ cd /var/www/html/detect
[ec2-user@ip-172-31-34-27 detect]$ pwd /var/www/html/detect
/var/www/html/detect
[ec2-user@ip-172-31-34-27 detect]$ ls
composer.json  composer.lock  index.php      sample.jpg      wwwroot
[ec2-user@ip-172-31-34-27 detect]$ sudo vim index.php
$hostname = "sample-app";

$sl = new SLCClient()
{
    username => "ec2-user-$?",
    password => "password",
    hostname => $host
};

key {
    $hostname.$sl->getObjId()
        => $objId,
        key => $keyname,
        publicKey => $pubKey,
        privateKey => $privKey;
}

if ($?) {
    $keypair = Get-KeyPair -Name $keyname;
    $imageId = Format-GUID($keypair.KeyId);
    if ($imageId) {
        $cmd = "aws ssm get-parameter --name /vsg/vm/$keypair.KeyId" -f $imageId

        $newKeyName = New-EkognitionClient {
            username => "ec2-user-$?",
            password => "password",
        };

        $result = $newKeyName->detectFaces();
        $keyname => "<>";
        $key => $keypair.PrivateKey;
        $keyid => $keypair.KeyId;
    }
}

```

4.upload success screenshot

```

[ec2-user@ip-172-31-34-37 ~]$ cat /var/www/html/detect
- Installing gpxml-rshttp/gpxr? (1.6.1): Downloading (100%)
- Installing gpxml-rshttp/gpxr? (1.6.2): Downloading (100%)
- Installing aws/aws-sdk-php (3.134.3): Downloading (100%)
gpxml-rshttp/gpxr suggests installing zendframework/zend-httpmodule/zend-httpmodule (Exit 2
88-T requires)
gpxml-rshttp/gpxr suggests installing gpx/loq (Required for using the log middle
ware)
gpxml-rshttp/gpxr suggests installing ext-intl (Required for Internationalized D
omain Name (IDN) support)
aws/aws-sdk-php suggests installing doctrine/cache (To use the DoctrineCacheAdap
ter)
aws/aws-sdk-php suggests installing aws/aws-php-aws-message-validator (To valida
te incoming SNS notifications)
Writing lock file
Generating autoload files
I package you are using is looking for funding.
See the packaged fund command to find out more!
[ec2-user@ip-172-31-34-37 detect]$ sudo wget https://i.pinimg.com/originals/69/7/
a/697ea316542c7494b804823b6c0c0180.jpg
--2020-04-23 15:04:08-- https://i.pinimg.com/originals/69/7a/697ea316542c74
94b804823b6c0c0180.jpg
Resolving i.pinimg.com (i.pinimg.com)... 101.101.146.64, 2a04:1e64:2::80:84
Connecting to i.pinimg.com (i.pinimg.com)|101.101.146.64|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 215551 (215K) [image/jpeg]
Saving to: '697ea316542c7494b804823b6c0c0180.jpg'

100%[=====] 215,551 --.-K/s in 0.00s

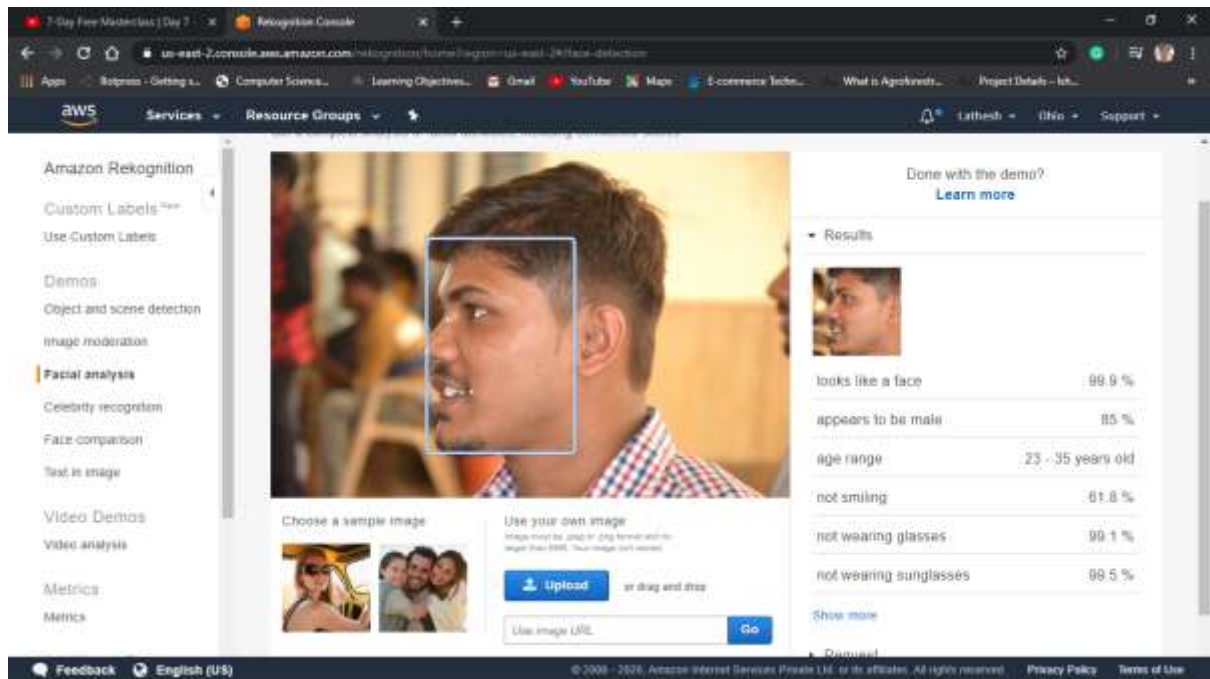
2020-04-23 15:04:08 (6.25 MB/s) - '697ea316542c7494b804823b6c0c0180.jpg' saved [
215551/215551]

[ec2-user@ip-172-31-34-37 detect]$ ls
697ea316542c7494b804823b6c0c0180.jpg composer.lock composer.lock vendor
[ec2-user@ip-172-31-34-37 detect]$ sudo mv 697ea316542c7494b804823b6c0c0180.jpg
sample.jpg
[ec2-user@ip-172-31-34-37 detect]$ ls
composer.lock composer.lock sample.jpg vendor
[ec2-user@ip-172-31-34-37 detect]$ sudo mv index.php
[ec2-user@ip-172-31-34-37 detect]$ ls
composer.lock composer.lock index.php sample.jpg vendor
[ec2-user@ip-172-31-34-37 detect]$ sudo index.php
sudo: index.php: command not found
[ec2-user@ip-172-31-34-37 detect]$ sudo php index.php
Image upload done... Here is the URL: https://www-lathash.s3-us-east-2.amazonaws
[ec2-user@ip-172-31-34-37 detect]$

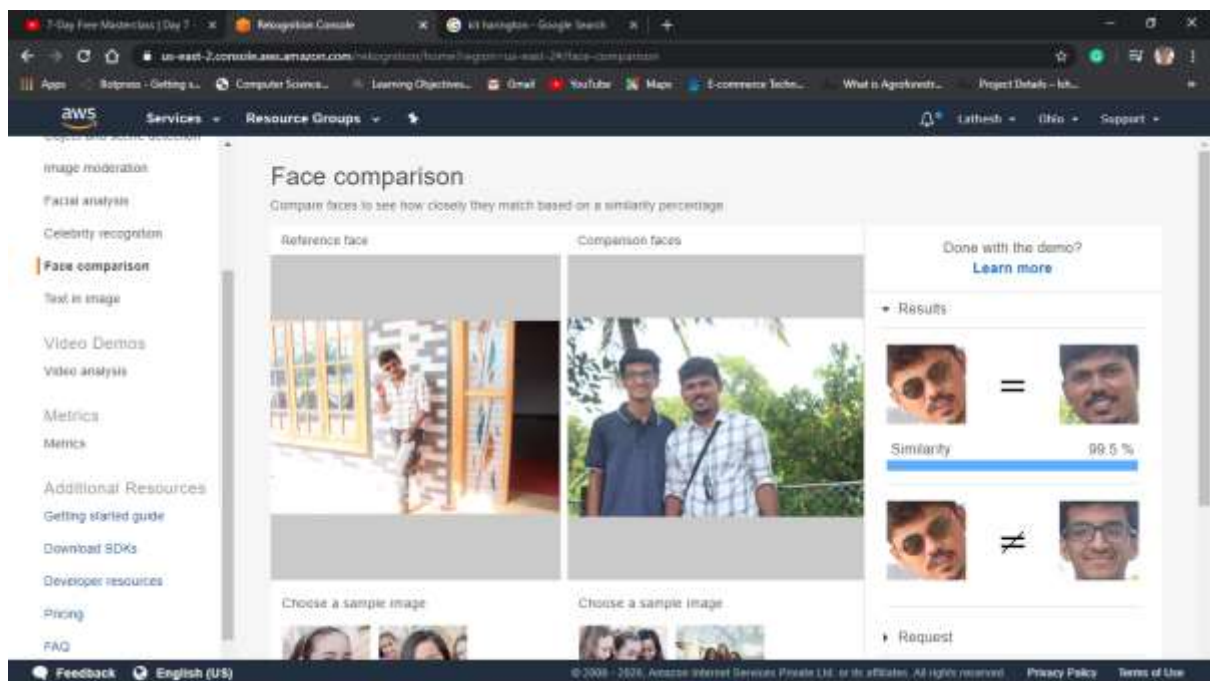
```


SCREENSHOTS FOR Rekognition

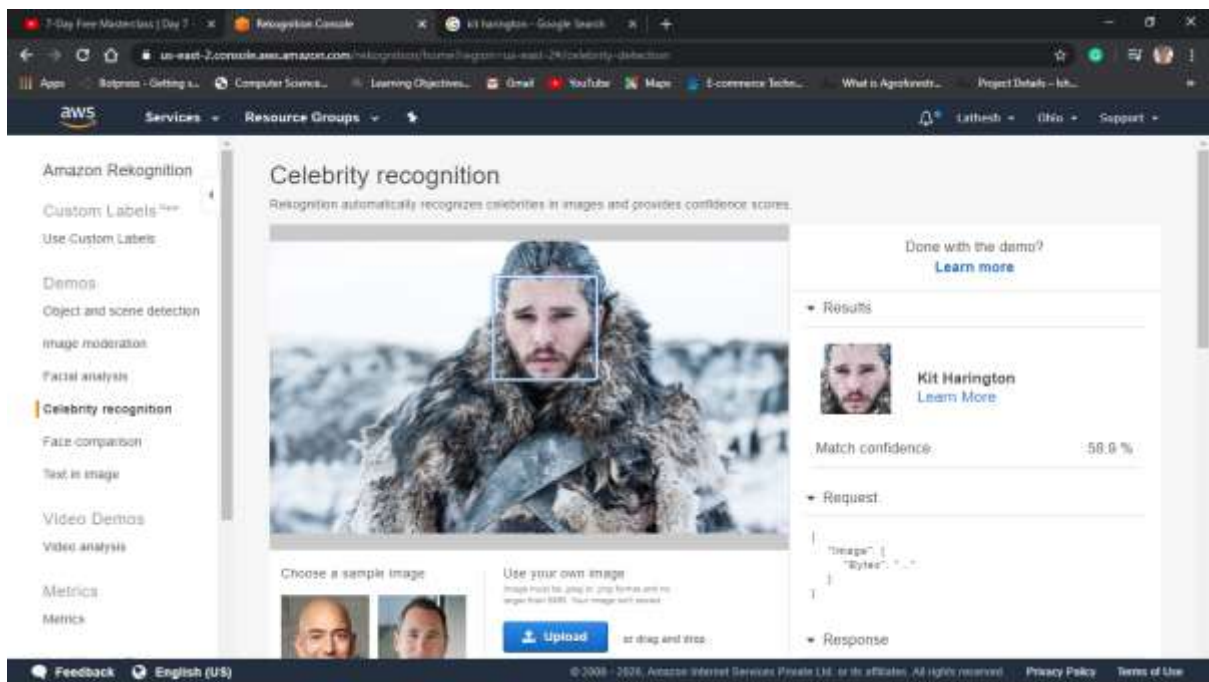
1.Face detect



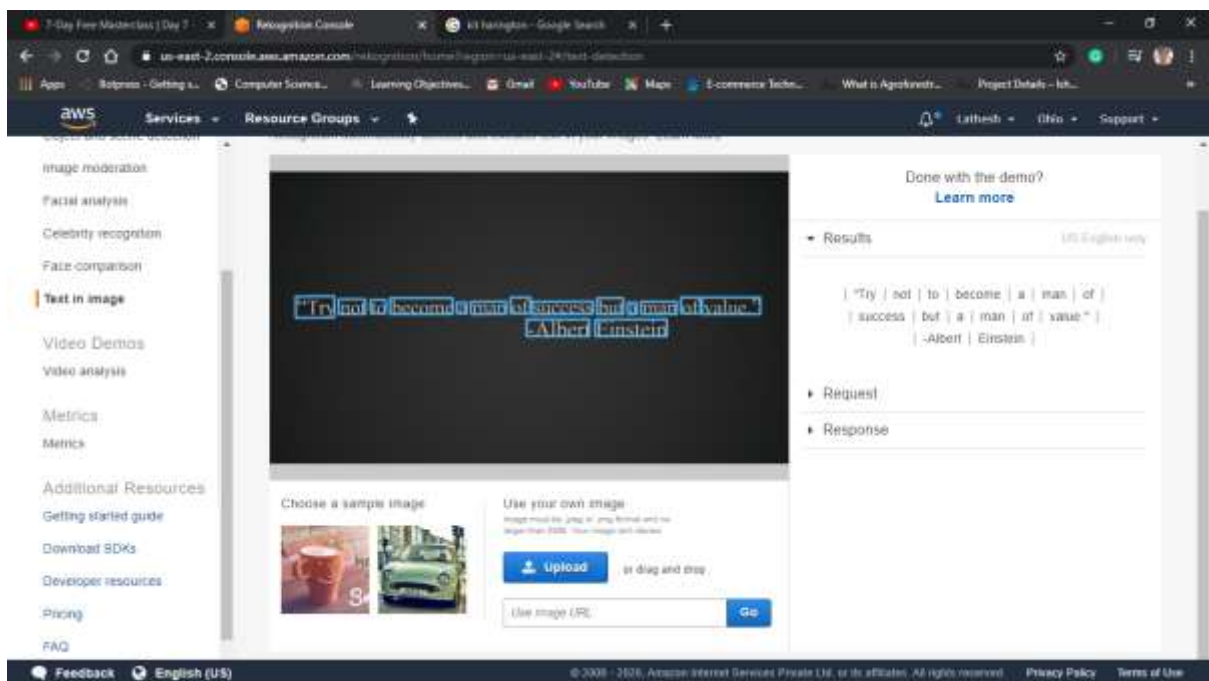
2.Face compare



3. Celebrity Rekognition



4. Text in image



1.Face detected success screenshot

