

ETHNUS AWS WEBINAR PROJECT

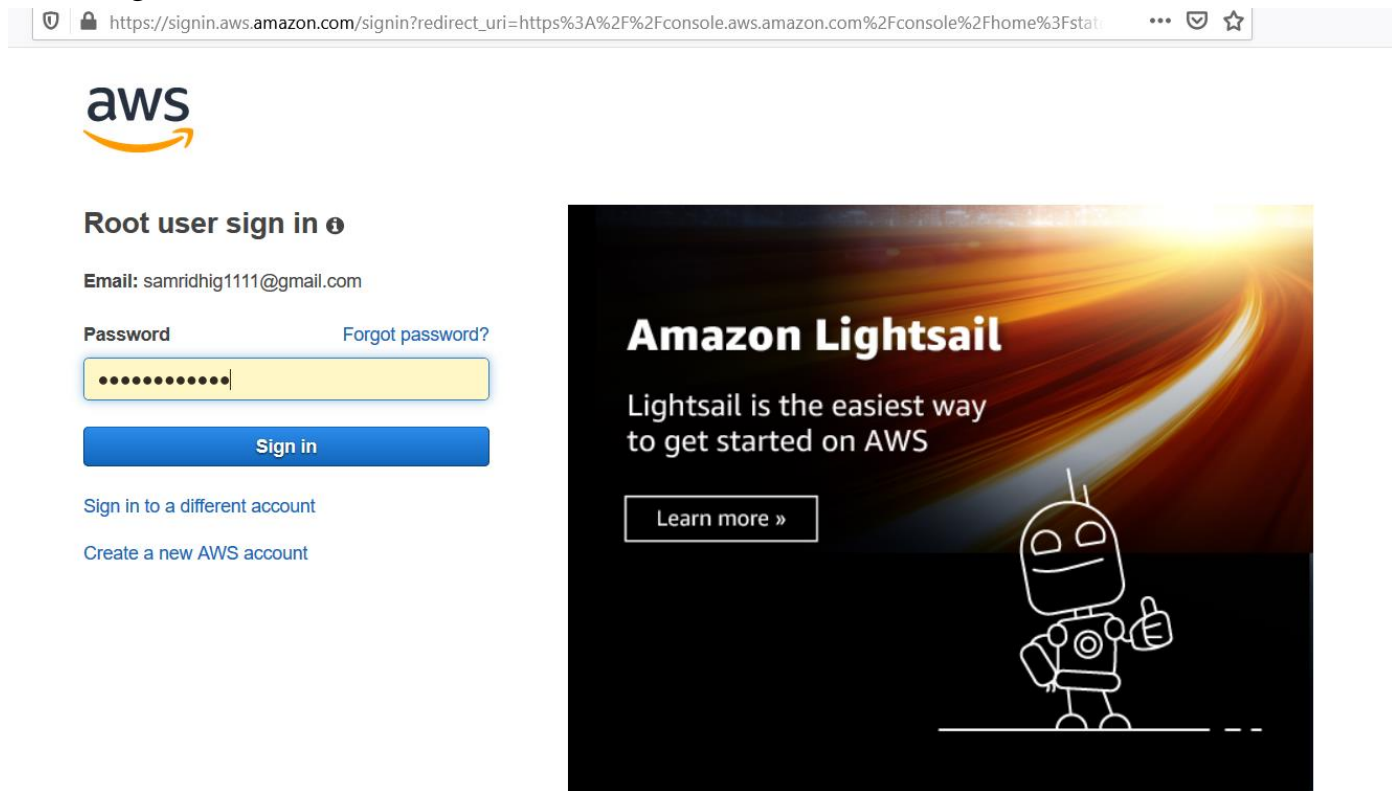
Submitted by- SAMRIDHI GUPTA

email id- samridhig1111@gmail.com

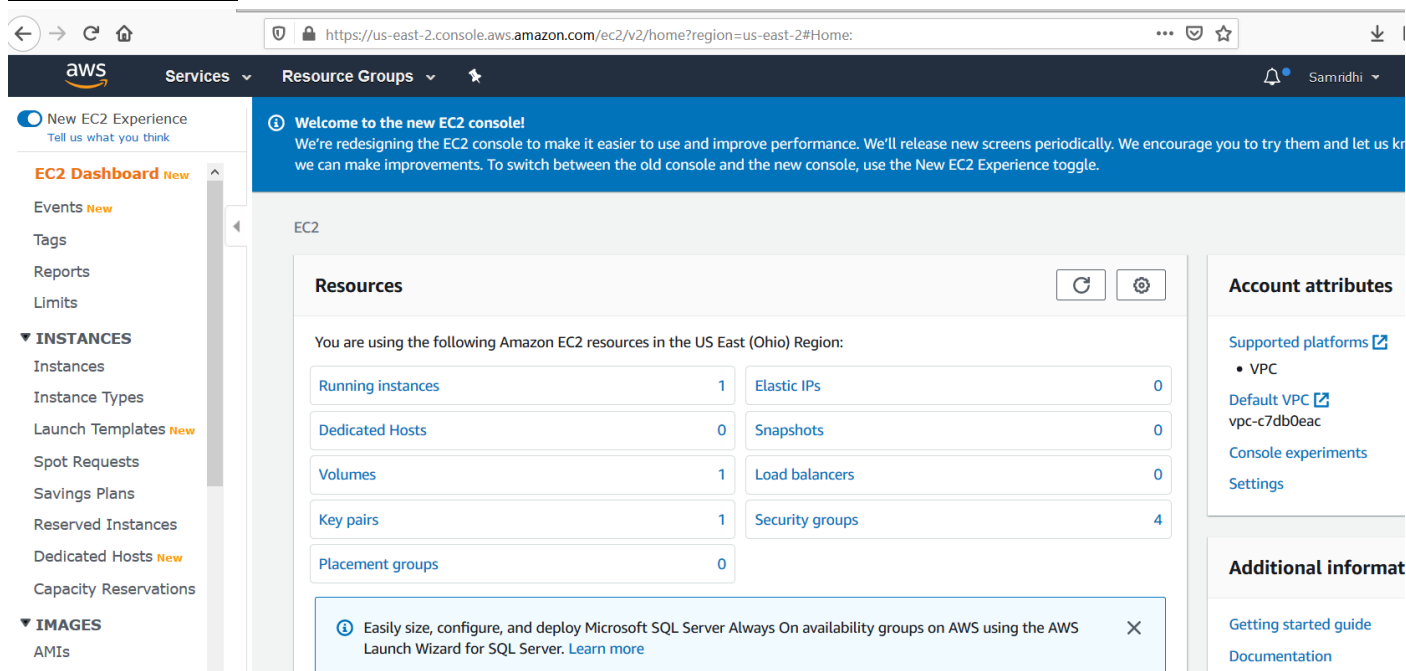
Screenshots-

Dashboards

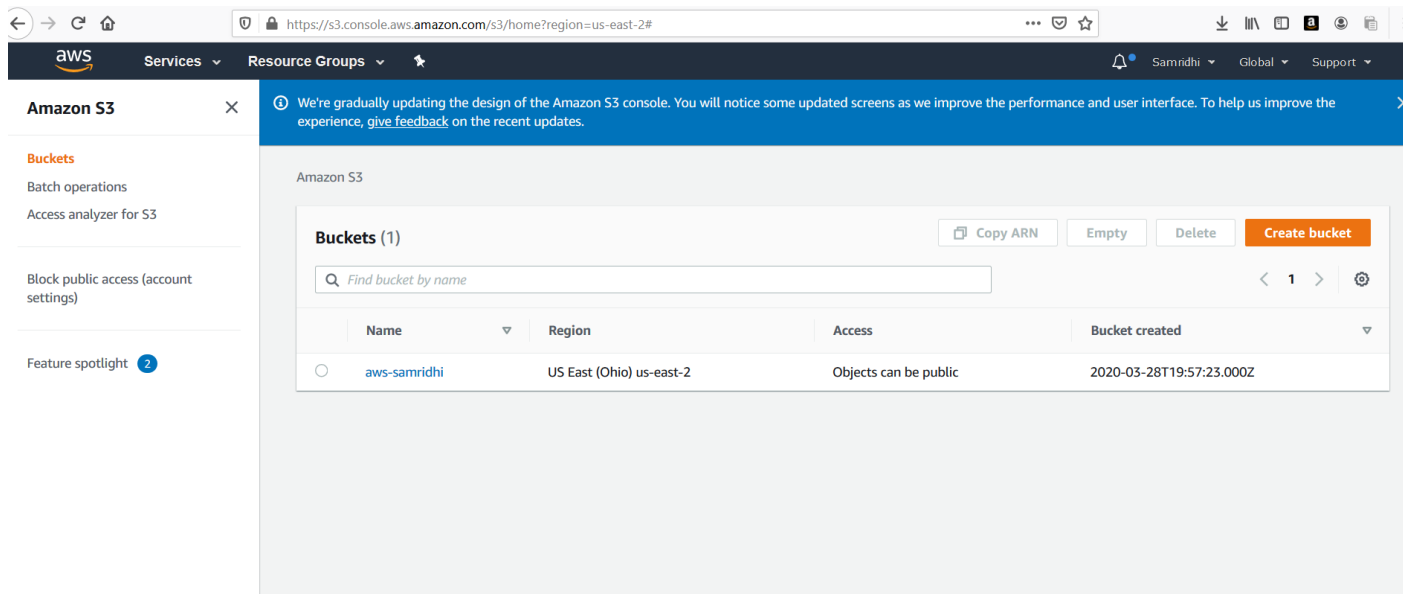
1. AWS login Screen



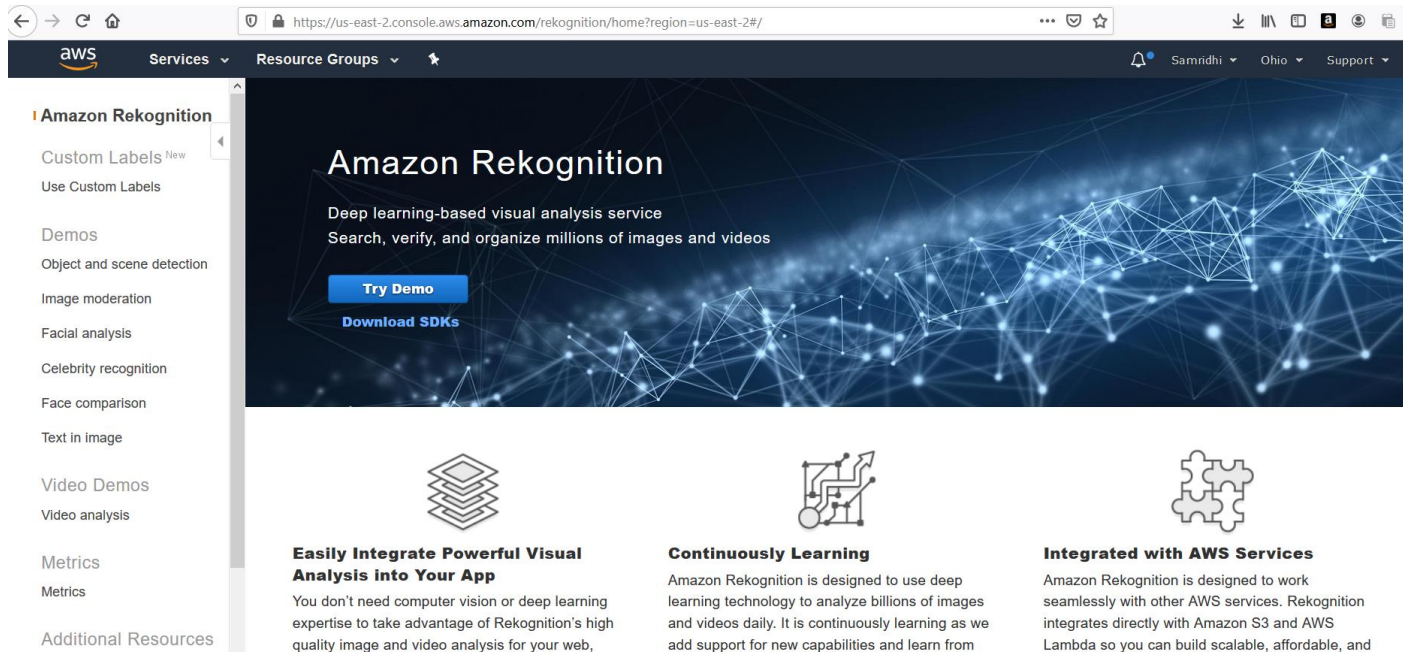
2. EC2 dashboard



3. S3 Dashboard



4. Rekognition Dashboard



EC2 Screenshots

1. Choosing an AMI

← → ↻ 🏠

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

⋮ 📄 ⭐

🔔 Samridhi Ohio Support

1. Choose AMI2. Choose Instance Type3. Configure Instance4. Add Storage5. Add Tags6. Configure Security Group7. Review

Step 1: Choose an Amazon Machine Image (AMI)

Cancel and Exit

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

Free tier eligible

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

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: ebsVirtualization type: hvmENA Enabled: Yes

64-bit (x86)64-bit (Arm)

Select

Amazon Linux

Free tier eligible

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: ebsVirtualization type: hvmENA Enabled: Yes

64-bit (x86)

Select

Red Hat

Free tier eligible

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: ebsVirtualization type: hvmENA Enabled: Yes

64-bit (x86)64-bit (Arm)

Select

2. Choosing an instance type

← → ↻ 🏠

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

⋮ 📄 ⭐

🔔 Samridhi Ohio Support

1. Choose AMI2. Choose Instance Type3. Configure Instance4. Add Storage5. Add Tags6. Configure Security Group7. Review

Step 2: Choose an Instance Type

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

Filter by: All instance typesCurrent generationShow/Hide Columns

	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
<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

3. Adding Storage

←→↺🏠

https://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 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.

4. Configuring Security Group

←→↺🏠

https://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:

Description:

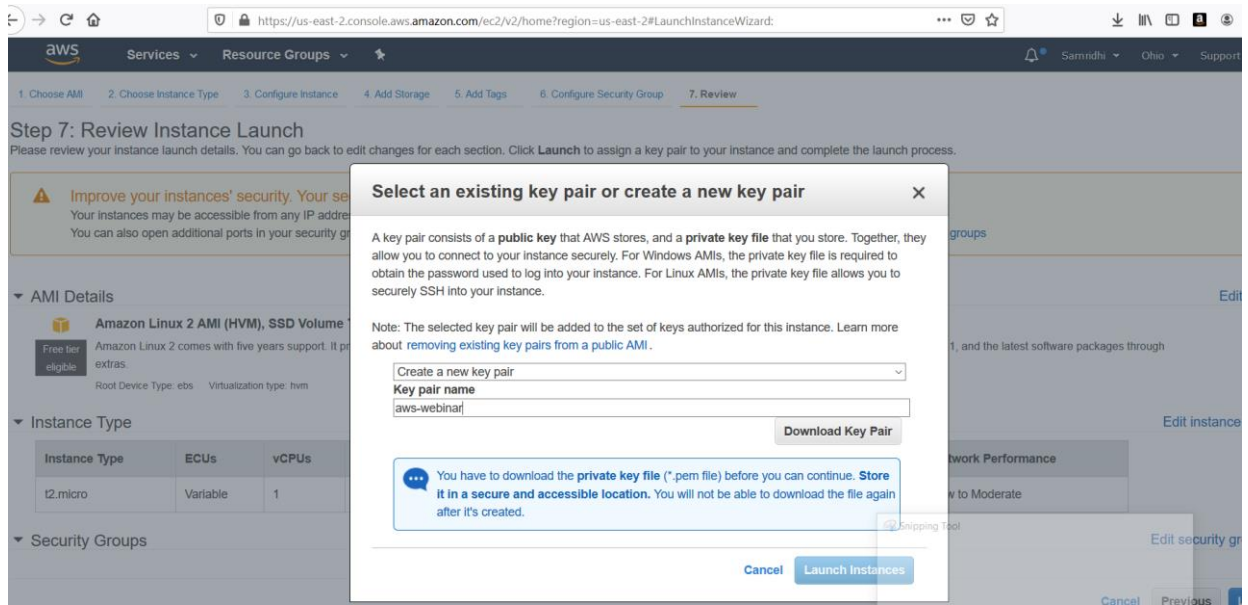
Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ	Description ⓘ
SSH ▾	TCP	22	Custom ▾ 0.0.0.0/0	e.g. SSH for Admin Desktop

Add Rule

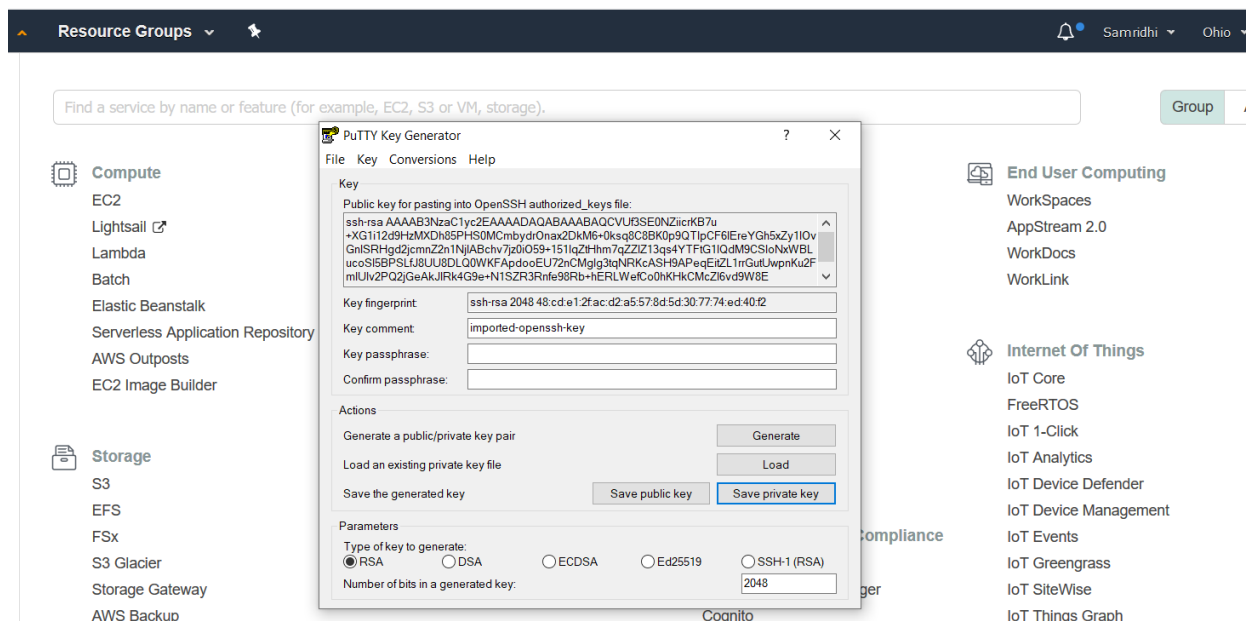
⚠️ Warning

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.

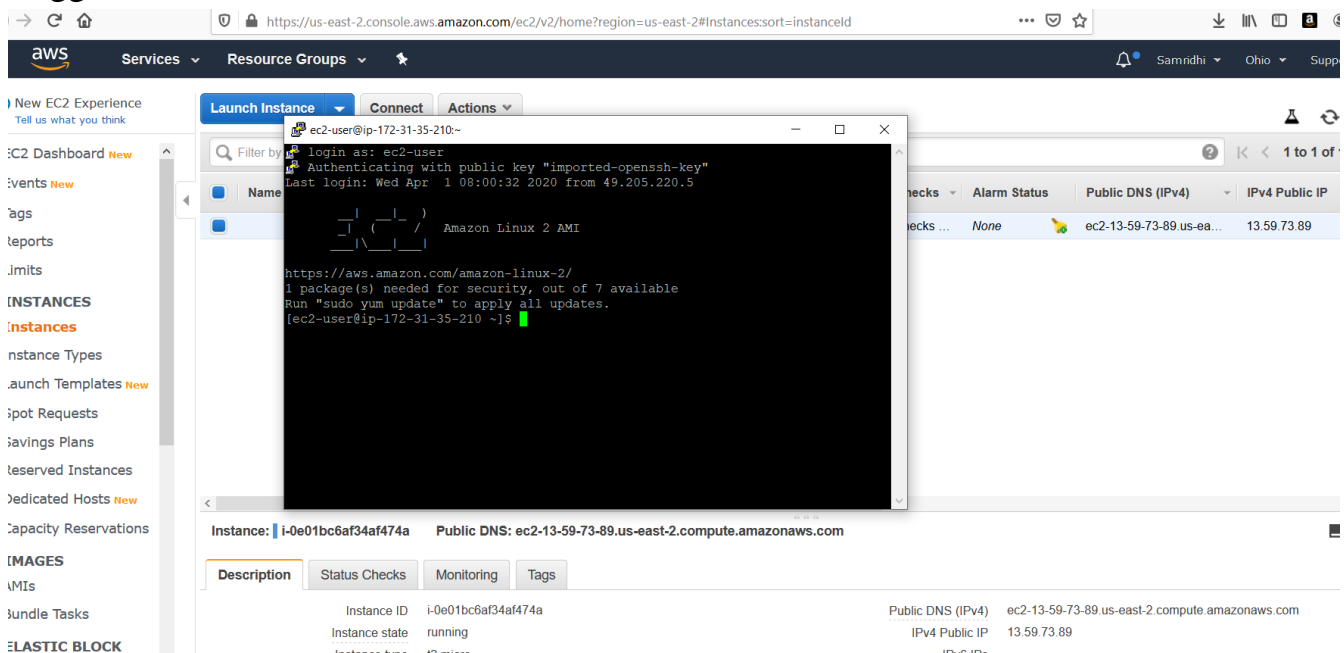
5. Key pair download



6. PEM to PPK

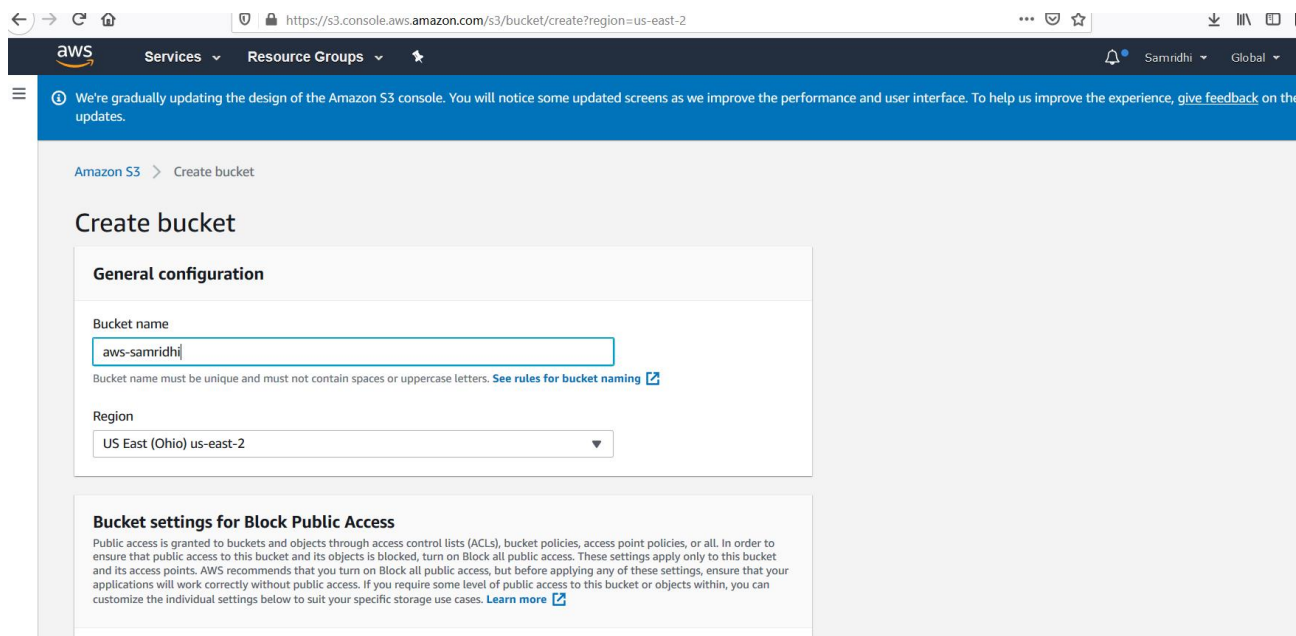


7. Logged into EC2

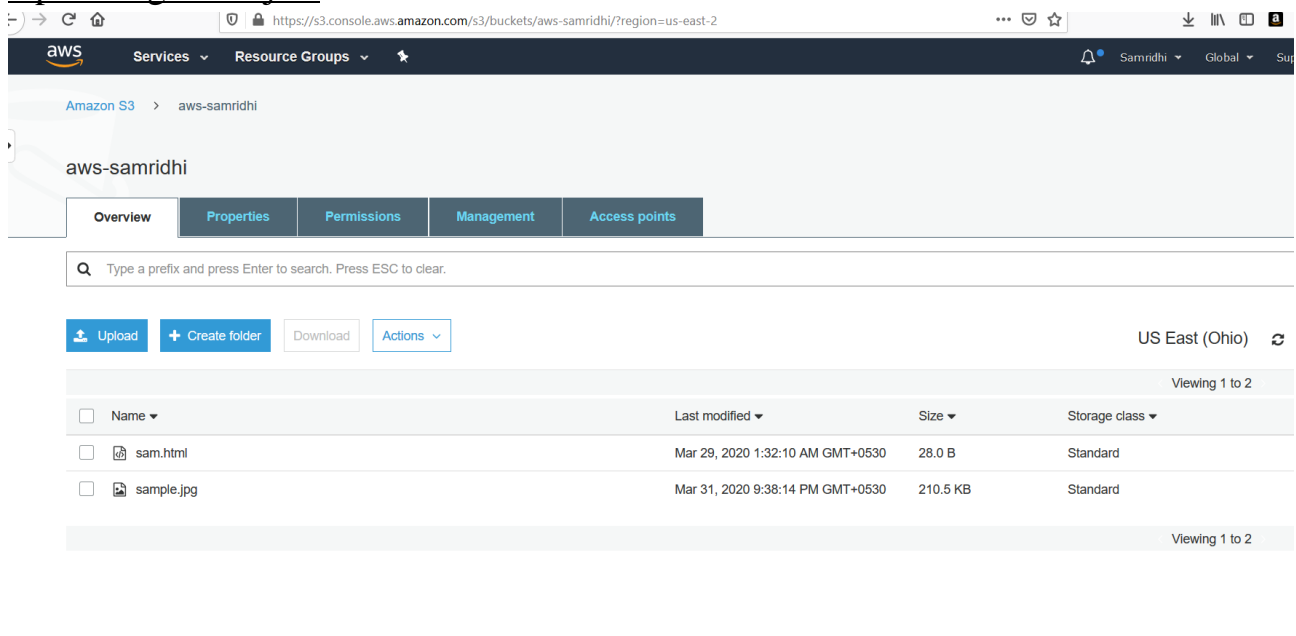


Screenshots for S3

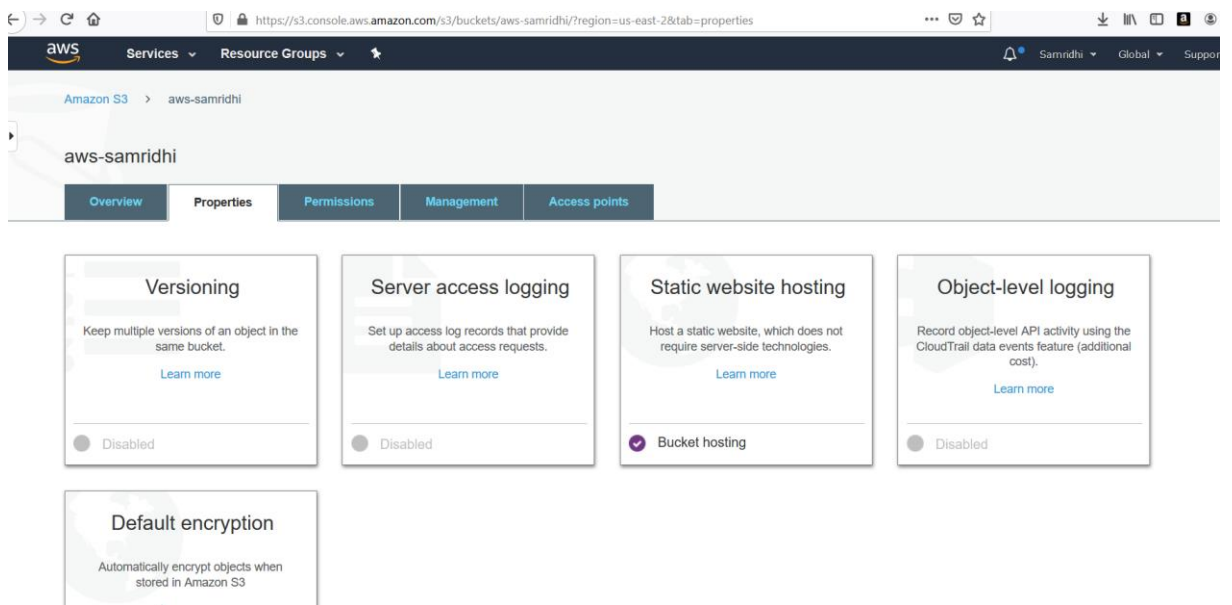
1. Creating a bucket



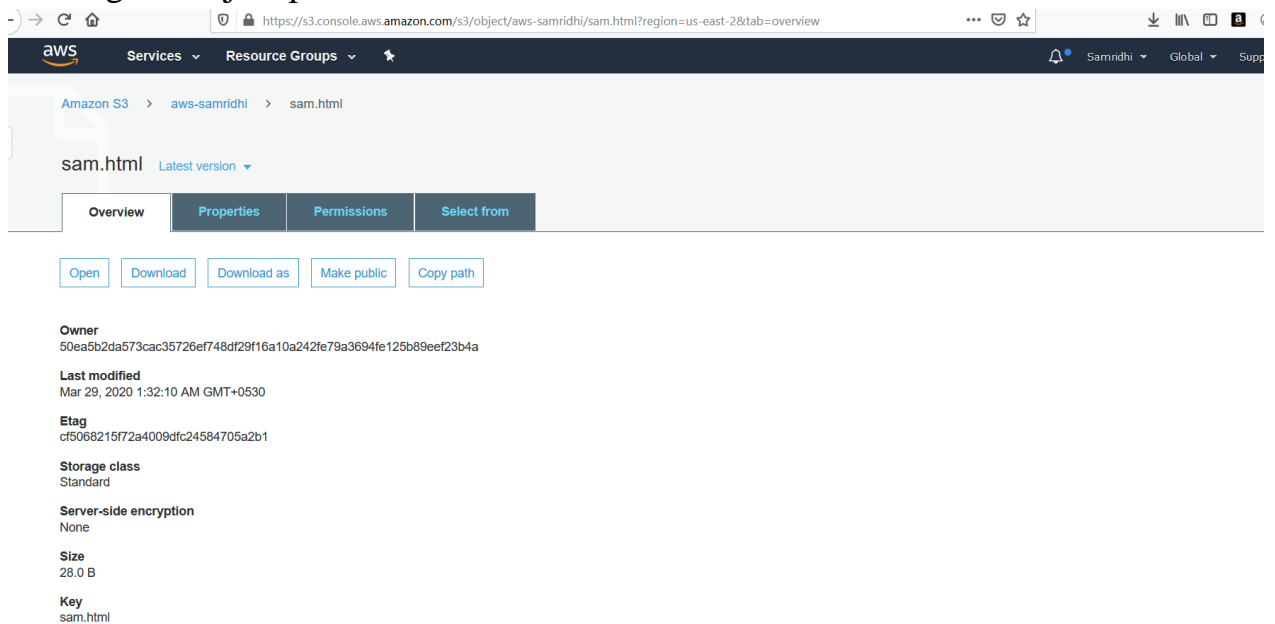
2. Uploading an Object



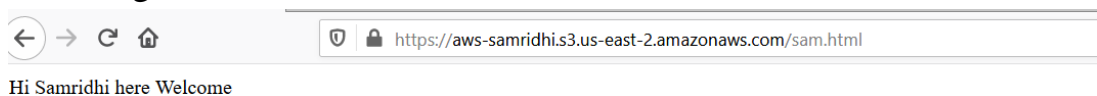
3. Enabling Static Website



4. Making the object public



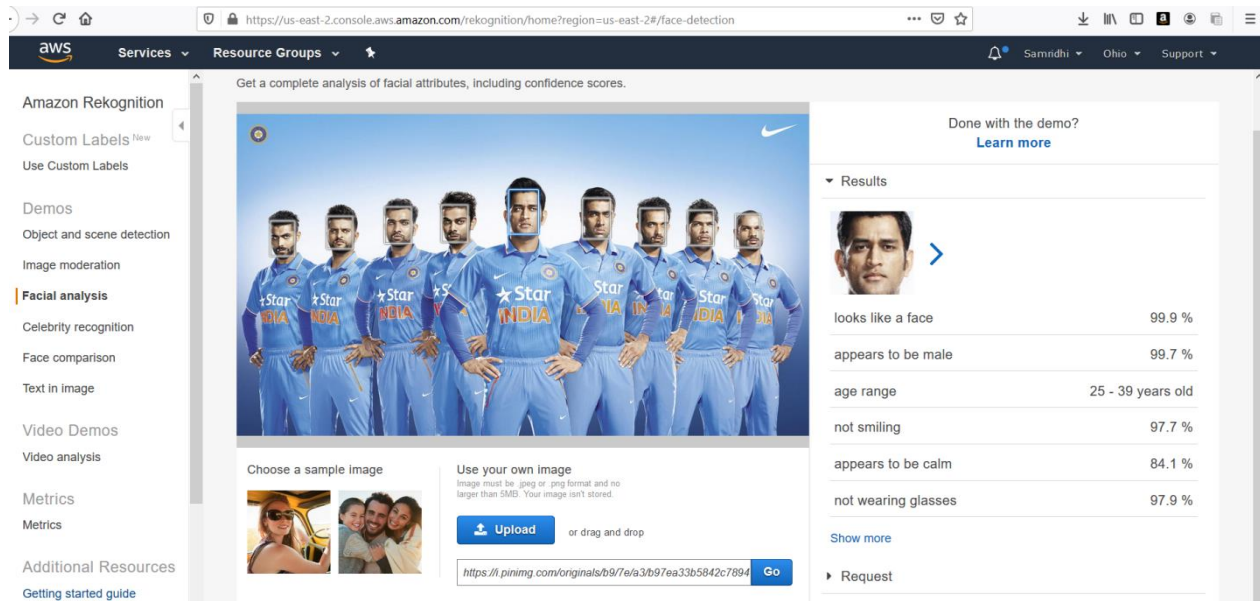
5. Checking the S3 link on browser



Rekognition

1. Face Detect

Get a complete analysis of facial attributes, including confidence scores.



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

▼ Results

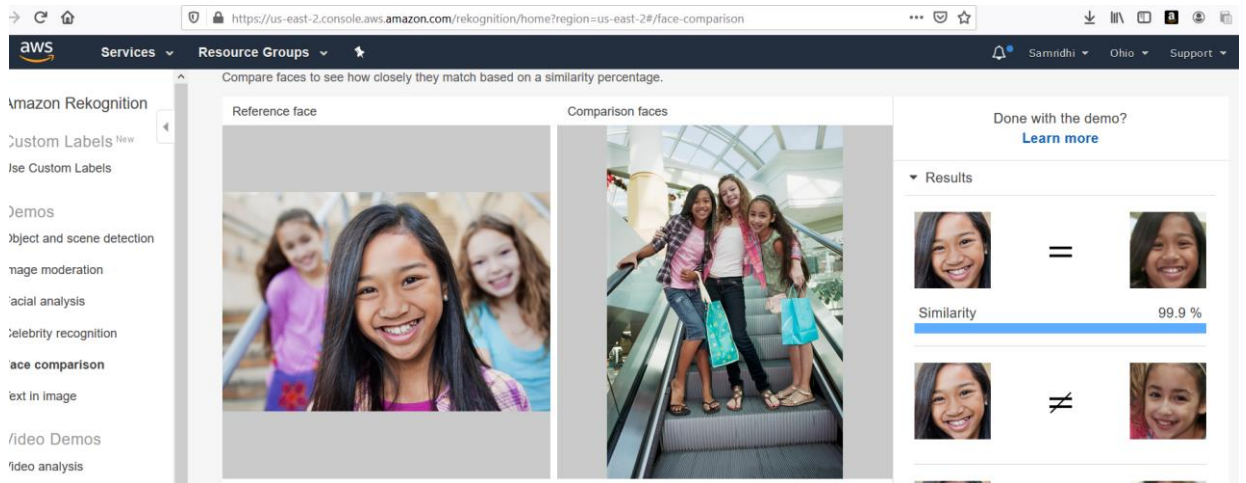
looks like a face	99.9 %
appears to be male	99.7 %
age range	25 - 39 years old
not smiling	97.7 %
appears to be calm	84.1 %
not wearing glasses	97.9 %

[Show more](#)

► Request





2. Face Compare

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



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

▼ Results

	=	
Similarity		99.9 %
	≠	

3. Celebrity Recognition


```

[ec2-user@ip-172-31-14-221:~]$ cd /var/www/html/face
[ec2-user@ip-172-31-14-221 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 GB) copied, 13.3551 s, 80.4 MB/s
[ec2-user@ip-172-31-14-221 face]$ sudo /sbin/mkswap /var/swap.1
mkswap: /var/swap.1: insecure permissions 0644, 0600 suggested.
Setting up swapspace version 1, size = 1024 MiB (107373728 bytes)
no label, UUID=84c19778-1b00-4f72-b601-6deaf03304f
[ec2-user@ip-172-31-14-221 face]$ sudo /sbin/swapon /var/swap.1
swapon: /var/swap.1: insecure permissions 0644, 0600 suggested.
[ec2-user@ip-172-31-14-221 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)
Package guzzle/guzzle is abandoned, you should avoid using it. Use guzzlehttp/guzzle instead.
Writing lock file
Generating autoload files
[ec2-user@ip-172-31-14-221 face]$ █

```

```
ec2-user@ip-172-31-14-221:/var/www/html/face
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/d3 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

#!/usr/bin/perl
error_reporting(0);

require_once(__DIR__ . '/vendor/autoload.php');

use Aws\S3\S3Client;
use Aws\Rekognition\RekognitionClient;

$bucket = 'aws-webinar-rohit';
$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;
}
```

4. Success Screenshot

```
ec2-user@ip-172-31-14-221:/var/www/html/face
login as: ec2-user
Authenticating with public key "imported-openssh-key"
Passphrase for key "imported-openssh-key":
Last login: Tue Mar 31 07:45:03 2020 from 157.35.248.116

 _ |   _ |   )
_| ( _|_ /    Amazon Linux 2 AMI
 _|\__|_|_|_|

https://aws.amazon.com/amazon-linux-2/
1 package(s) needed for security, out of 7 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-14-221 ~]$ cd /var/www/html
[ec2-user@ip-172-31-14-221 html]$ cd face
[ec2-user@ip-172-31-14-221 face]$ sudo php index.php
Image upload done... Here is the URL: https://aws-webinar-rohit.s3.us-east-2.amazonaws.com/sample.jpg[ec2-user@ip-172-31-14-221 face]$ ^C
[ec2-user@ip-172-31-14-221 face]$
```

EC2 and Rekognition

1. Face Detect Success Screenshot

```
ec2-user@ip-172-31-14-221:/var/www/html/face
[ec2-user@ip-172-31-14-221 face]$ sudo php index.php
Image upload done... Here is the URL: https://aws-webinar-rohit.s3.us-east-2.amazonaws.com/sample.jpgTotally there are 9 faces[ec2-user@ip-172-31-14-221 face]$
```