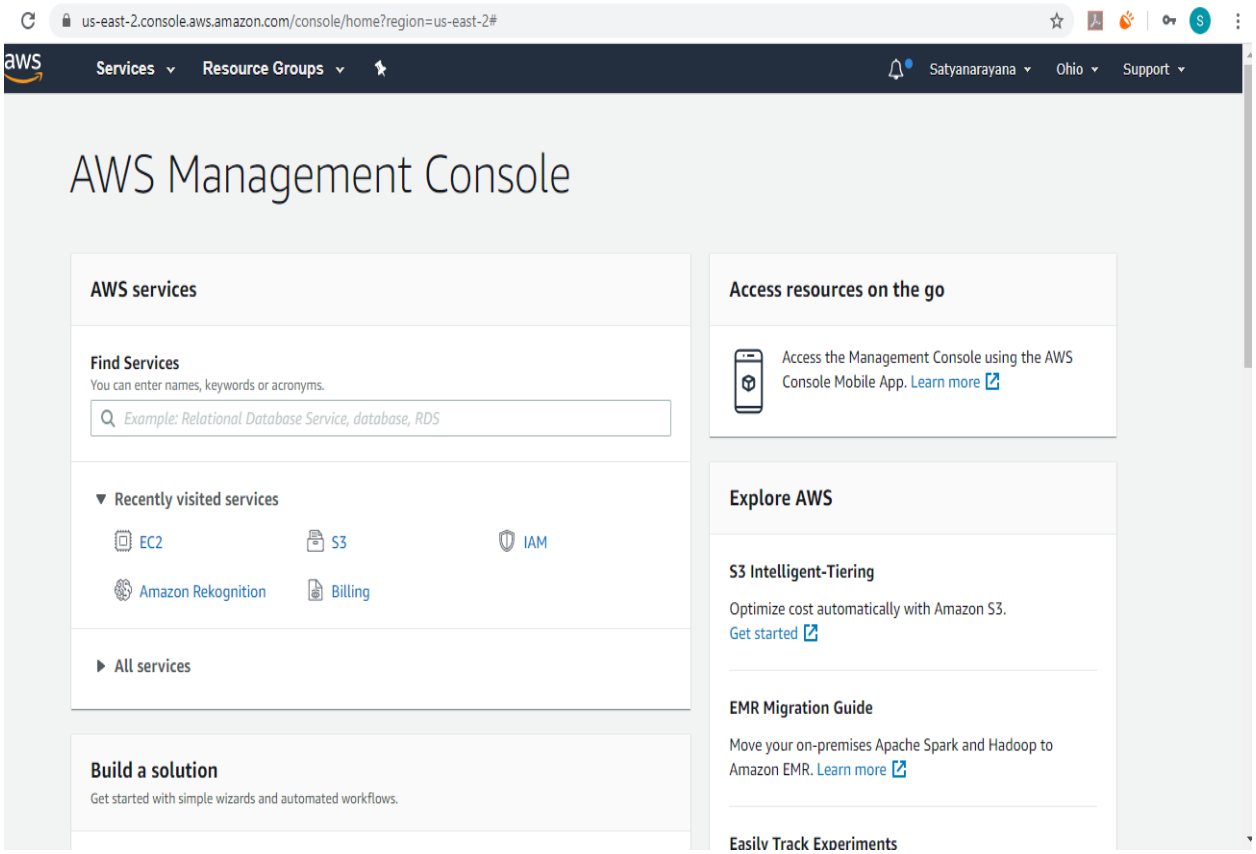
Name: Savarnika Chitti

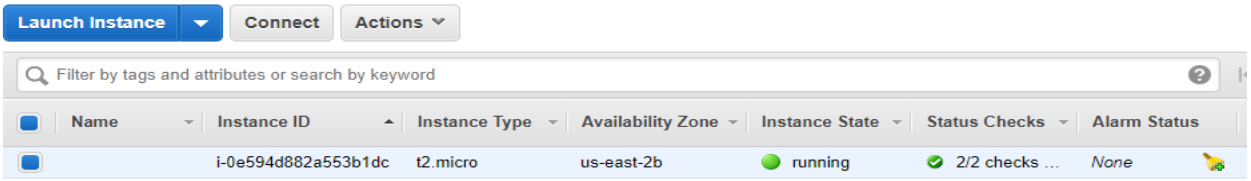
Clg: Vellore Institute of Technology

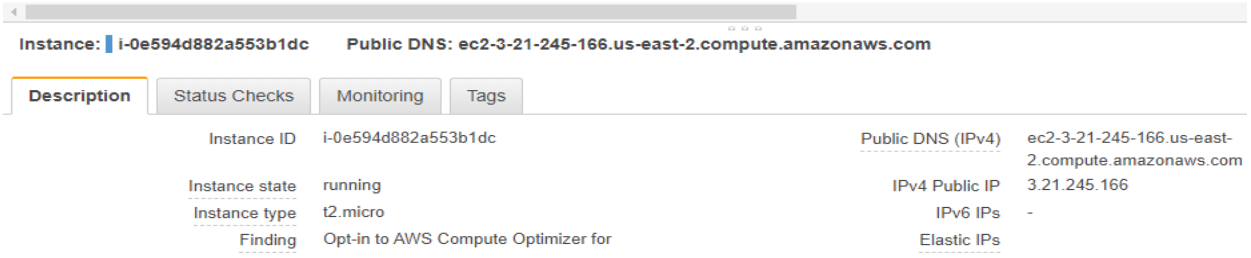
Screenshots for Dasboard

1. AWS login screen with username

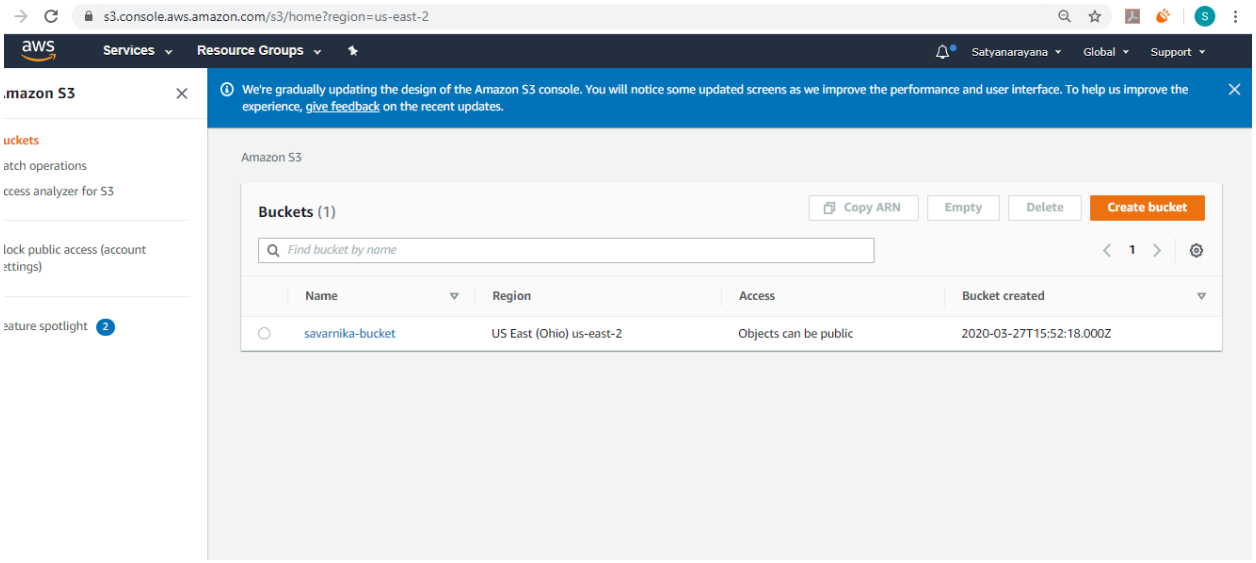


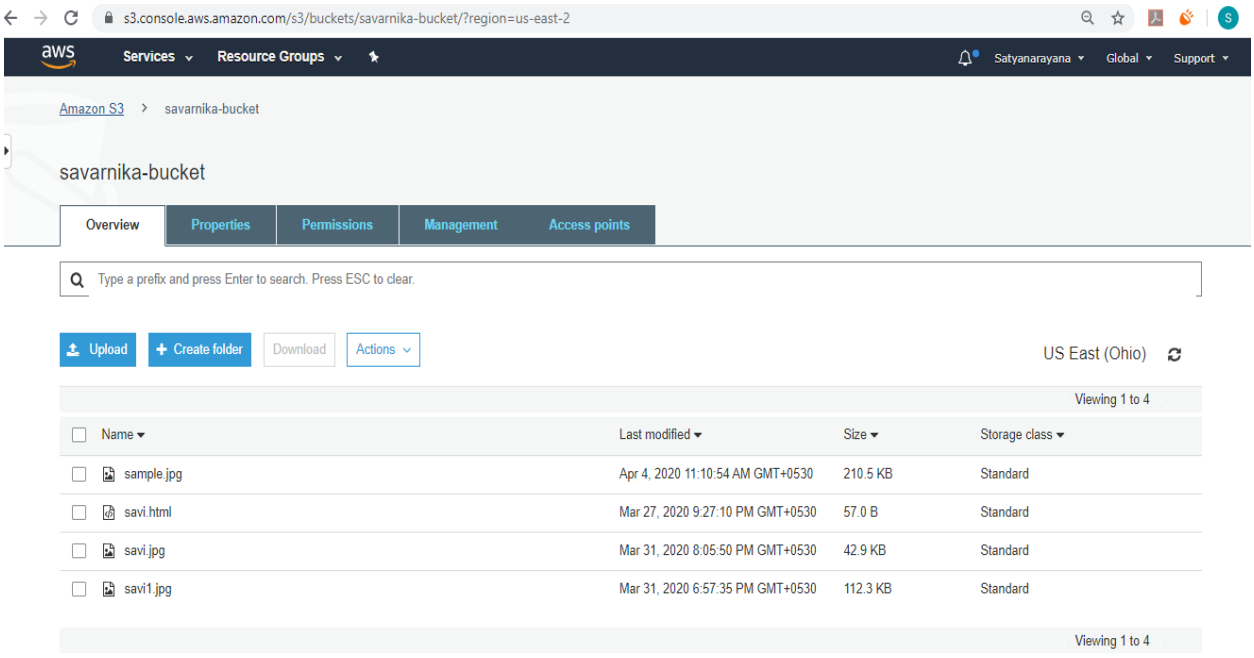

2. EC2 Dashboard



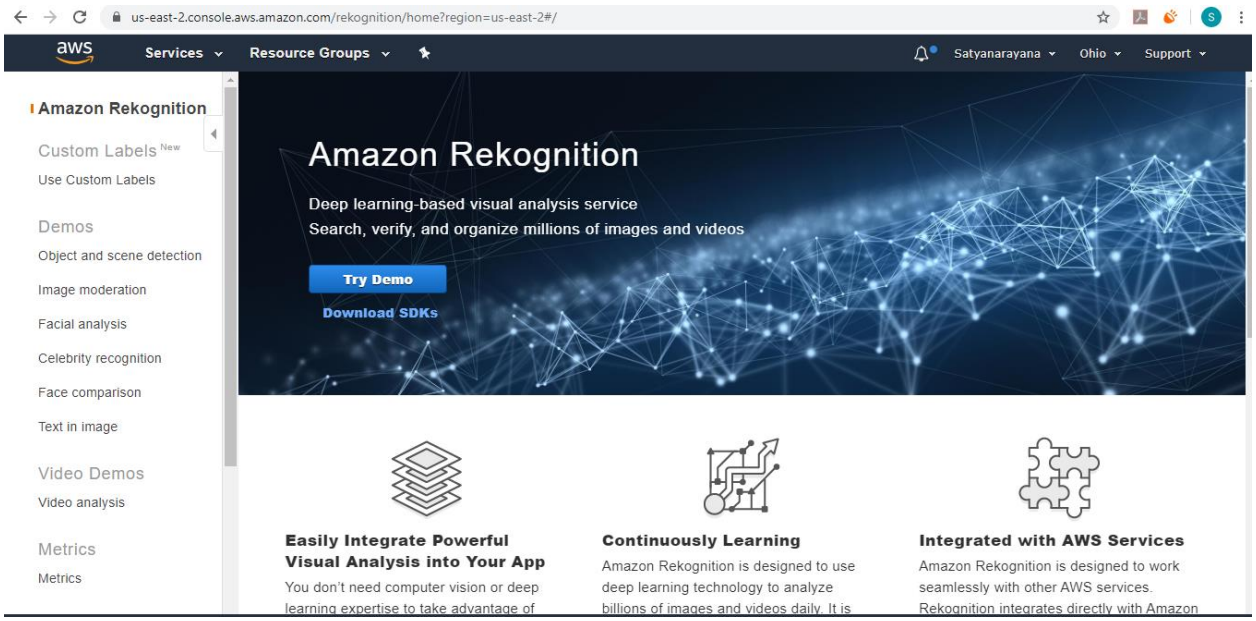




## 3. S3 Dashboard



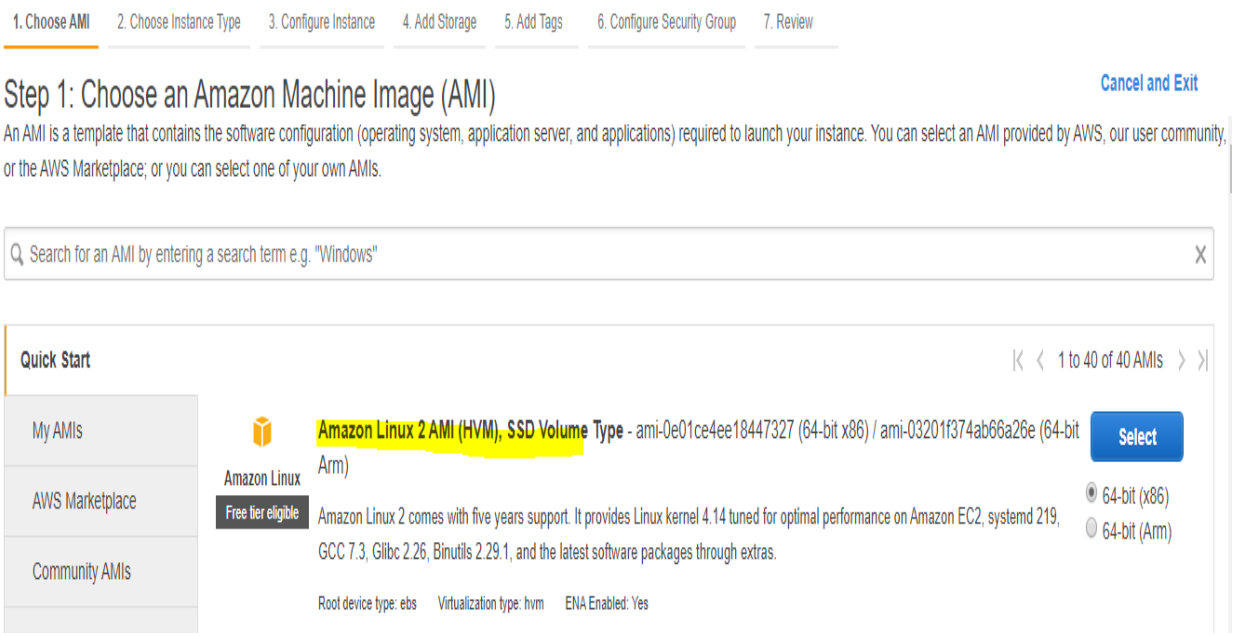



4. Rekognition Dashboard

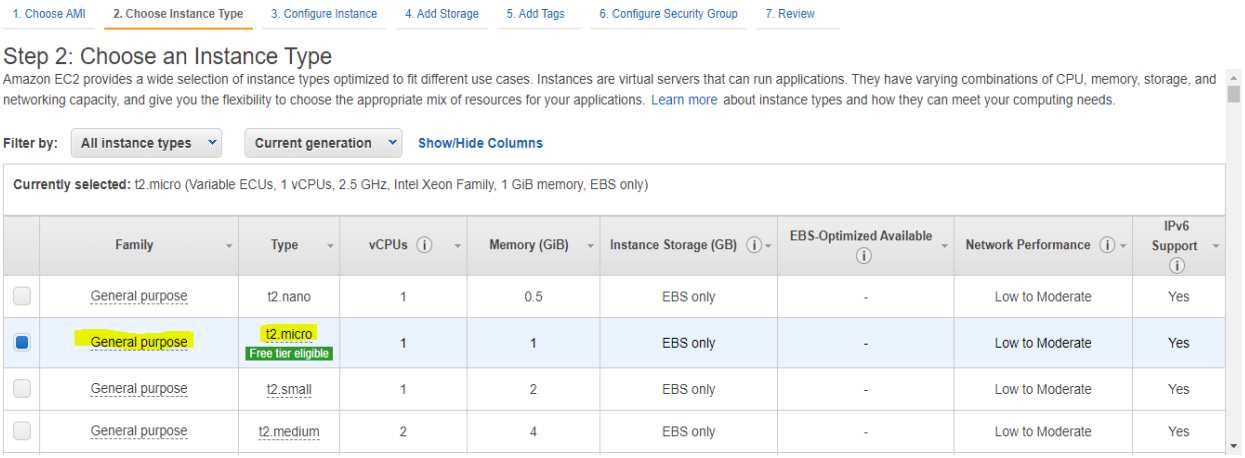



>> Screenshots for EC2

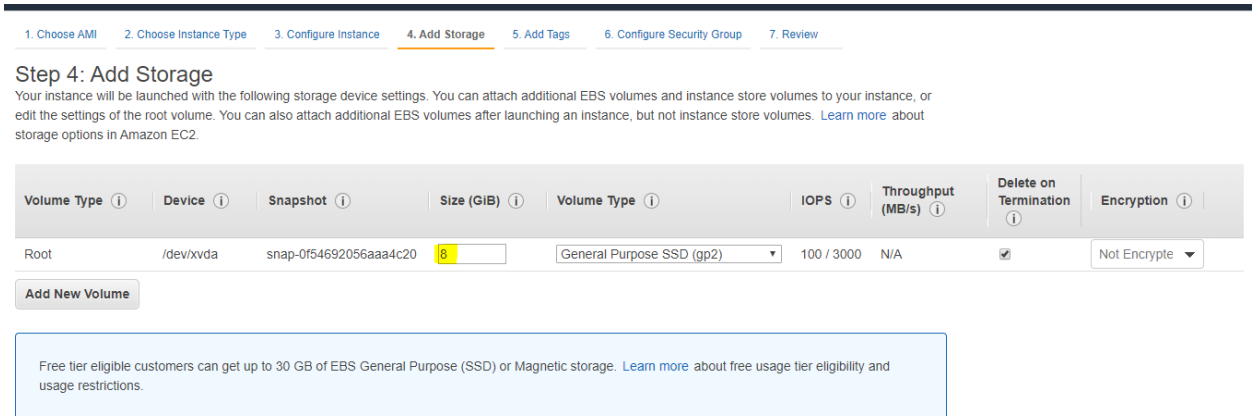
1. Choosing an AMI



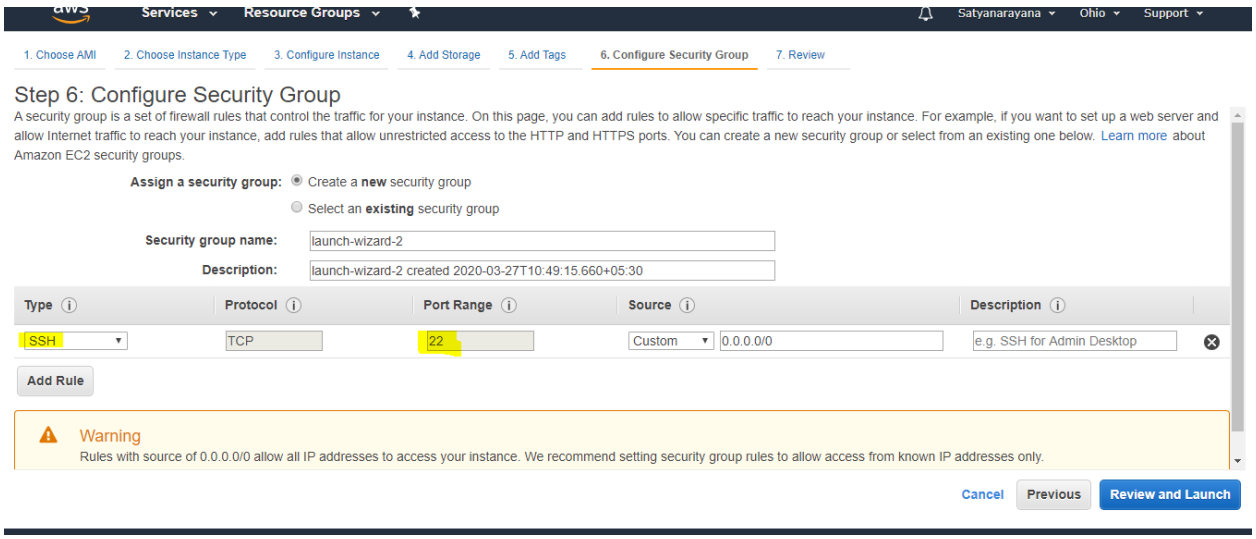

## 2. Choosing an Instance type




## 3. Adding Storage

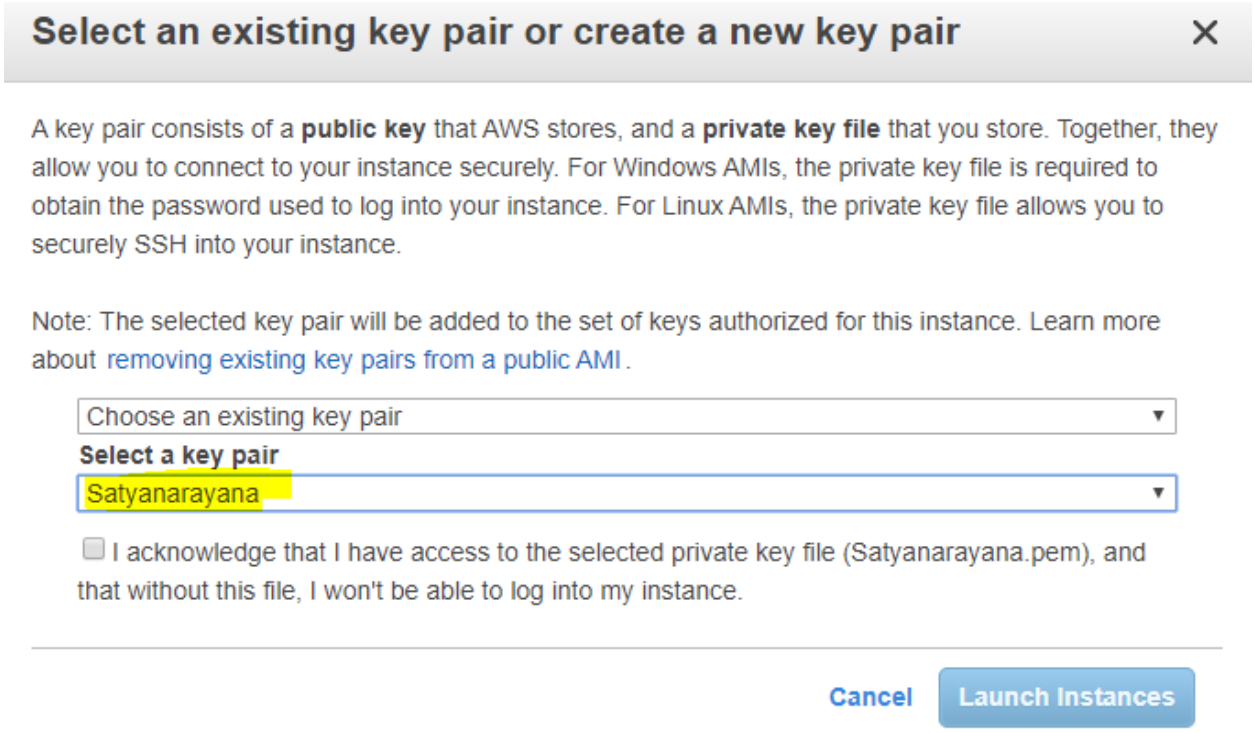



4. Configure Security group

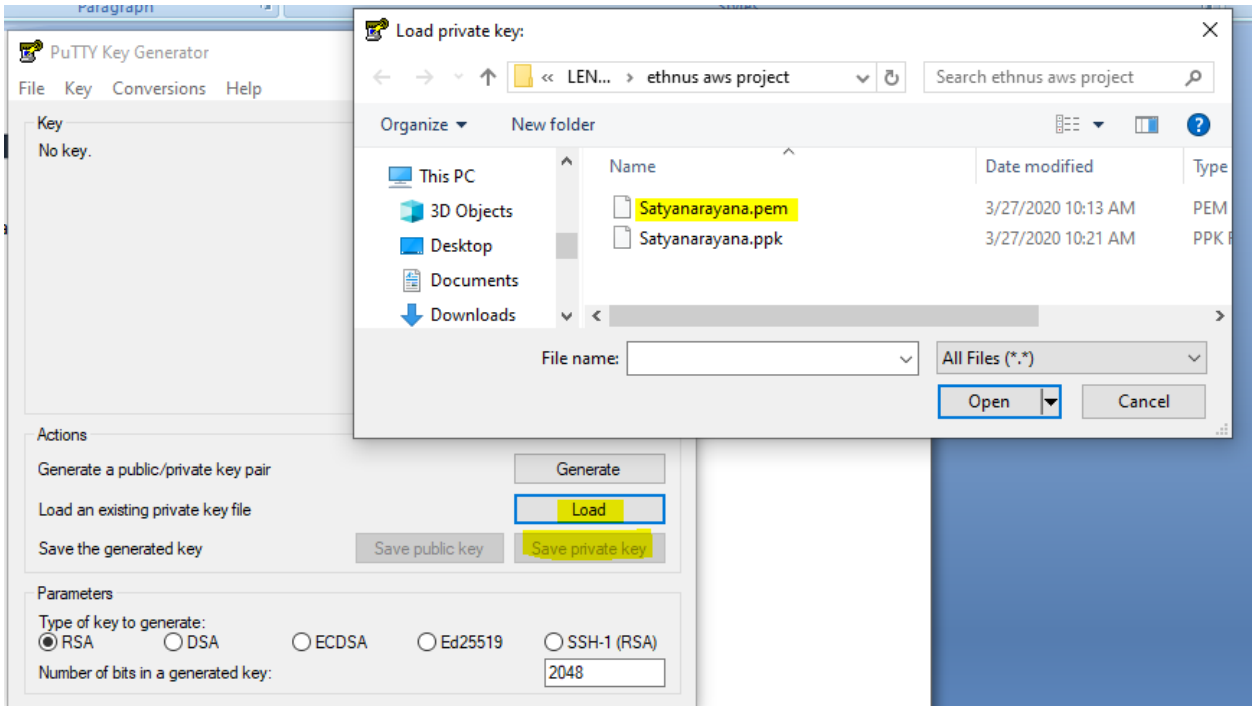



5. Key pair download

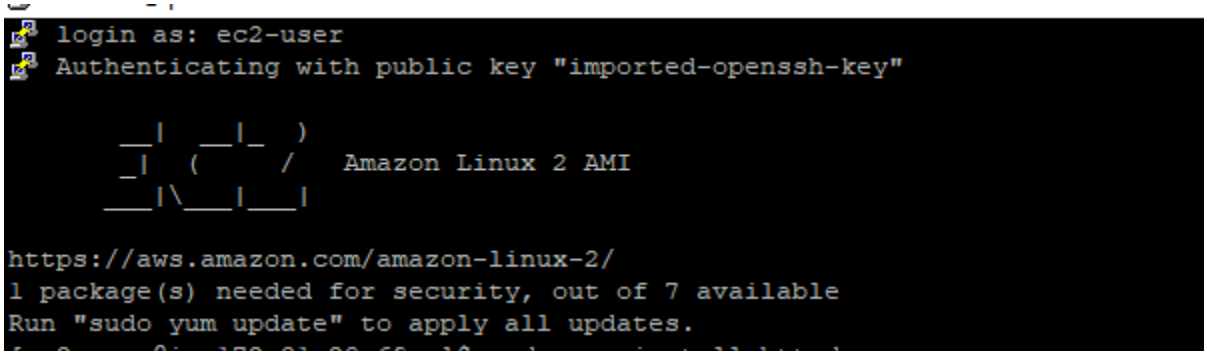
I did not take the screenshot while downloading I can provide the image while using the ppk file




6. PuTTYgen conversion from pem to ppk

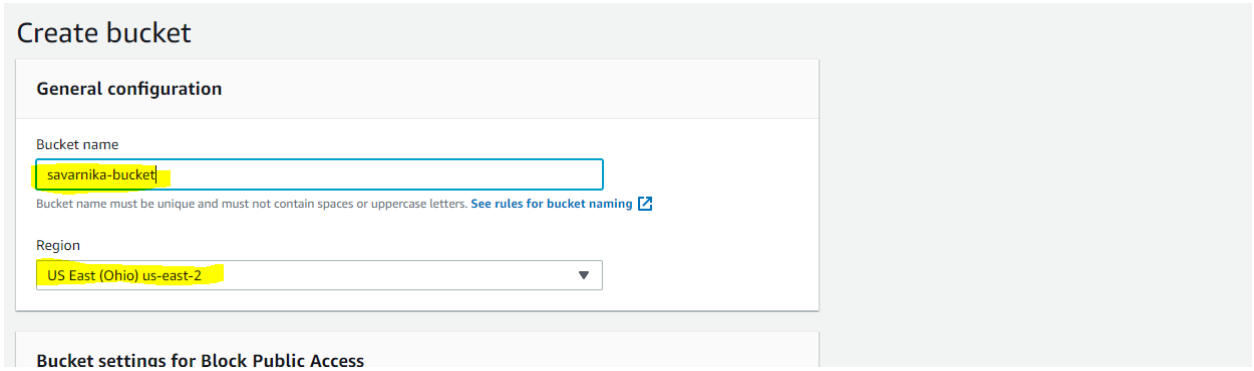


7. Logged in EC2 black screen

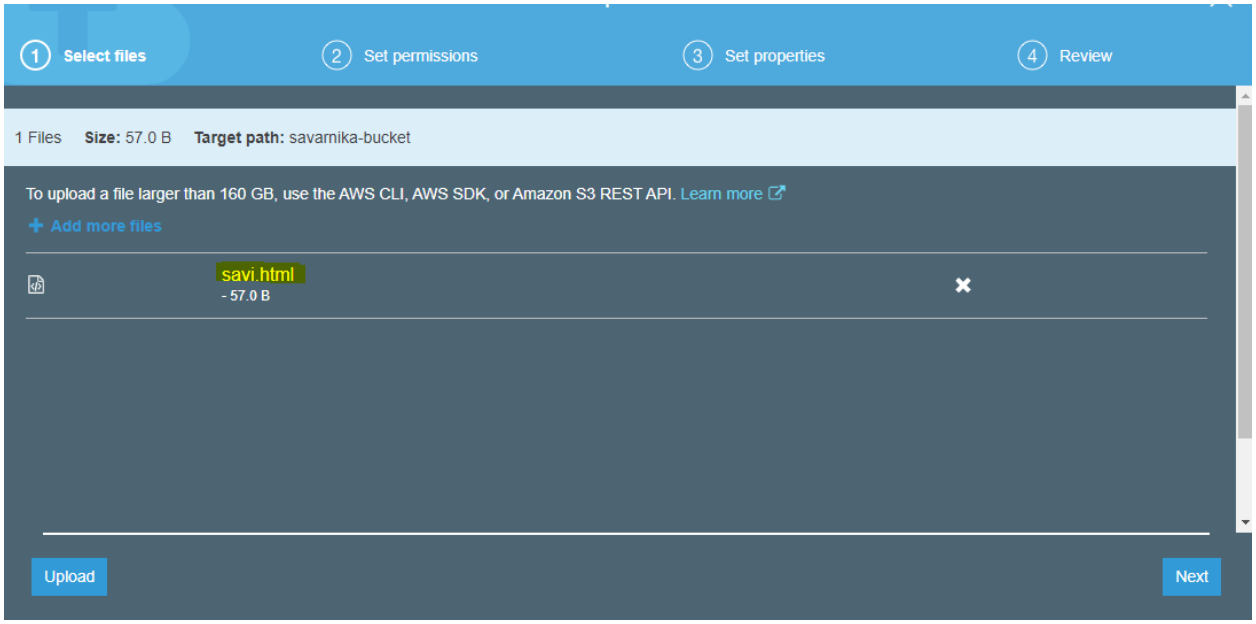


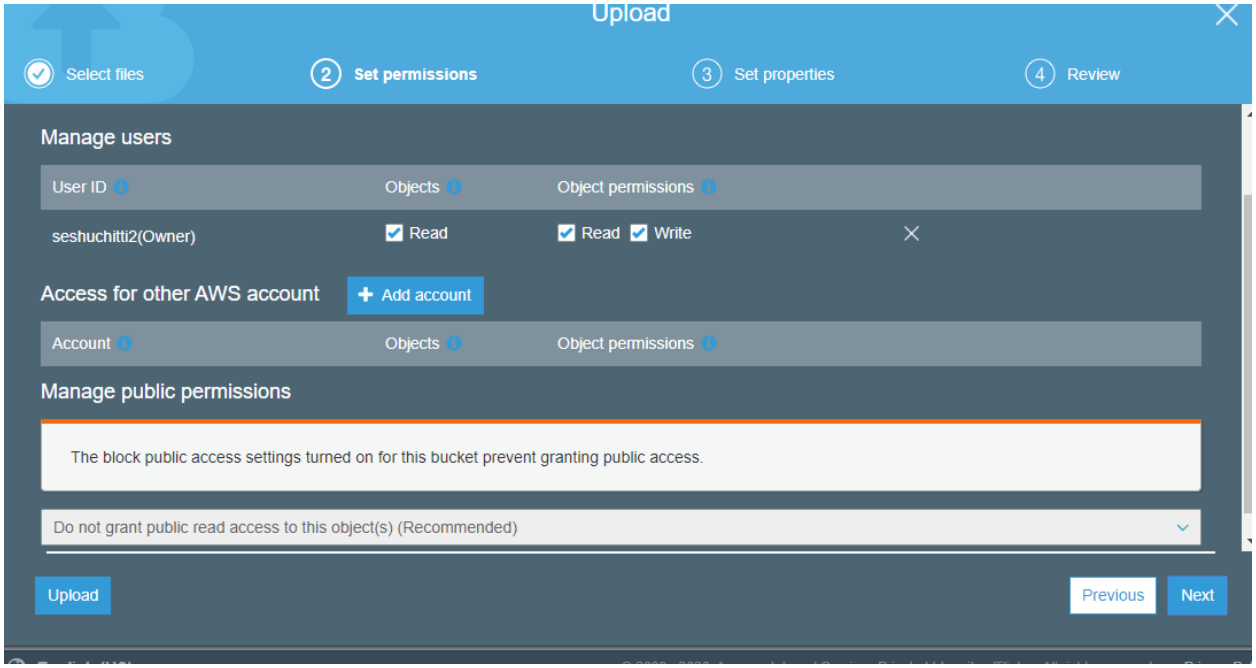

>>Screenshots for S3

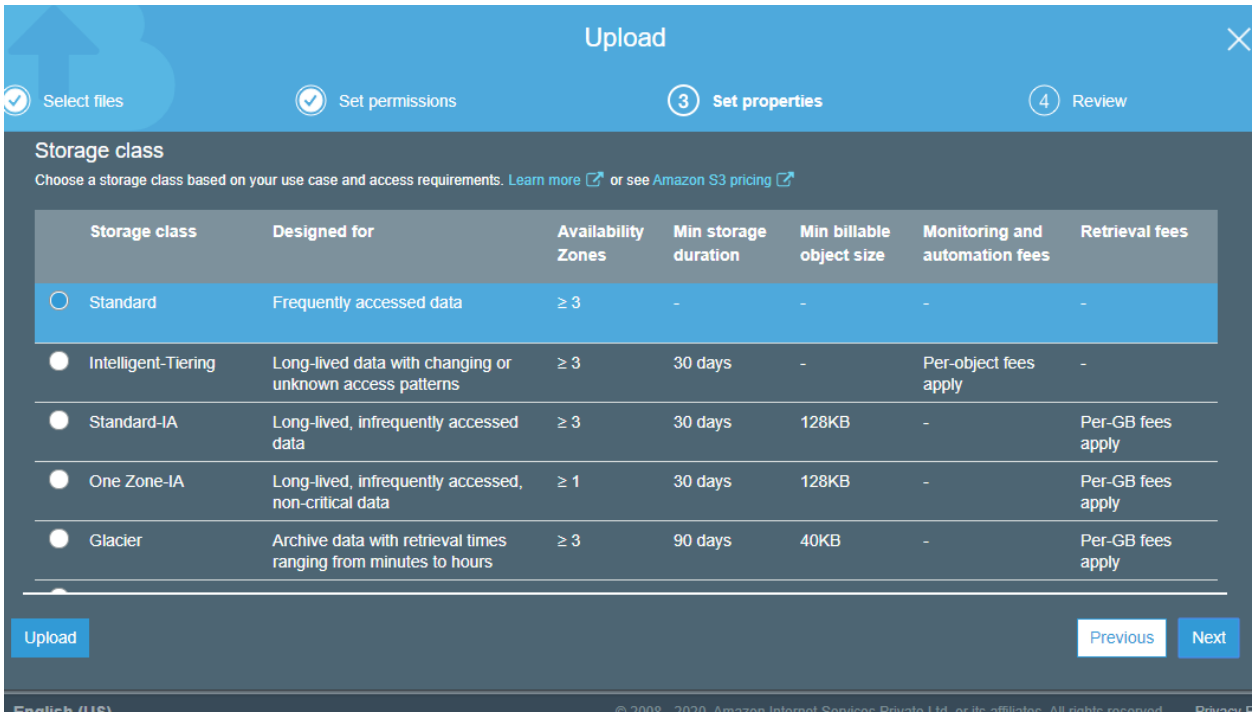
1. Creating a bucket

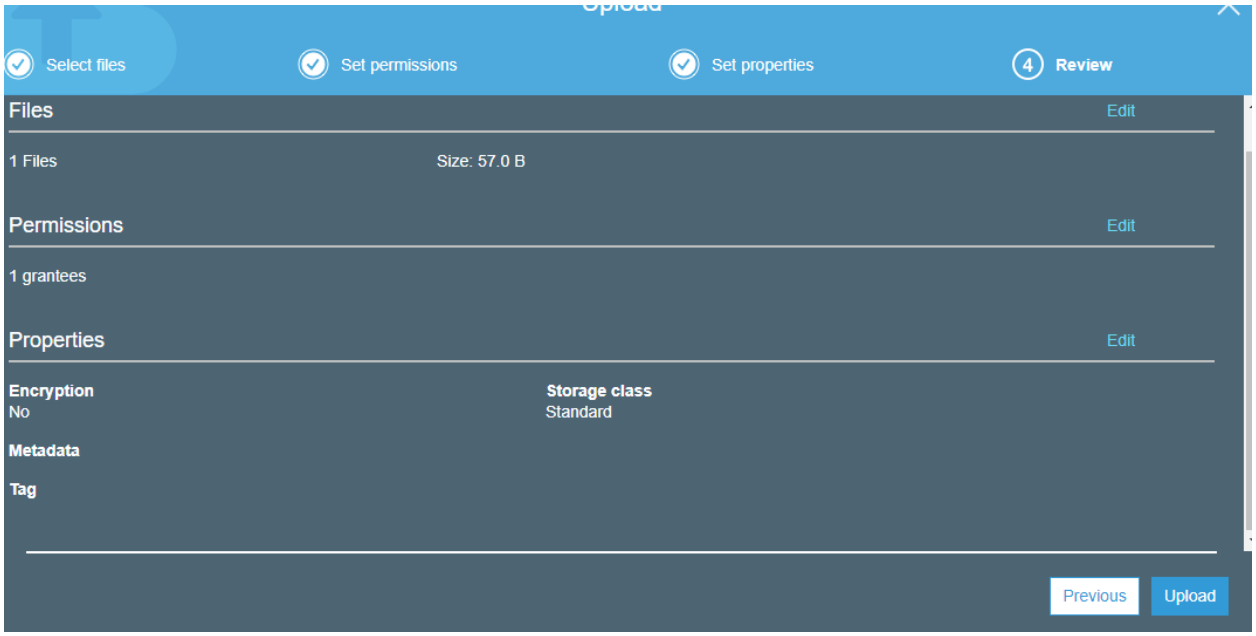


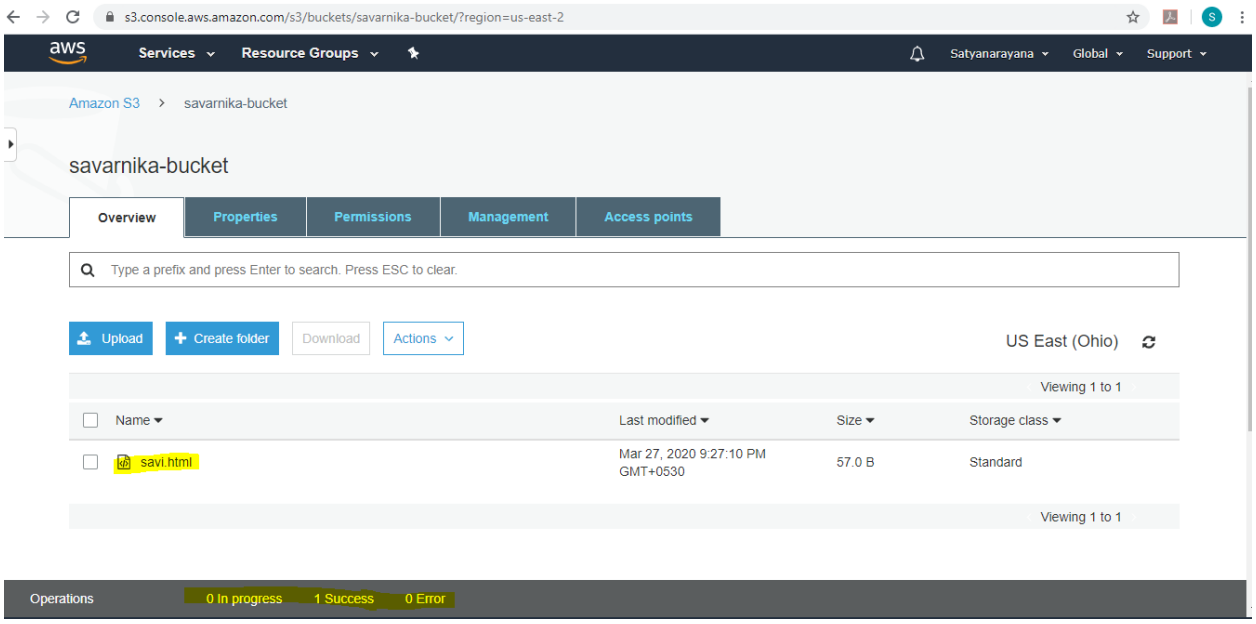

2. Uploading an Object


1 Select files
2 Set permissions
3 Set properties
4 Review
1 Files Size: 57.0 B Target path: savarnika-bucket
To upload a file larger than 160 GB, use the AWS CLI, AWS SDK, or Amazon S3 REST API. Learn more
Add more files
savi.html
- 57.0 B
Upload
Next

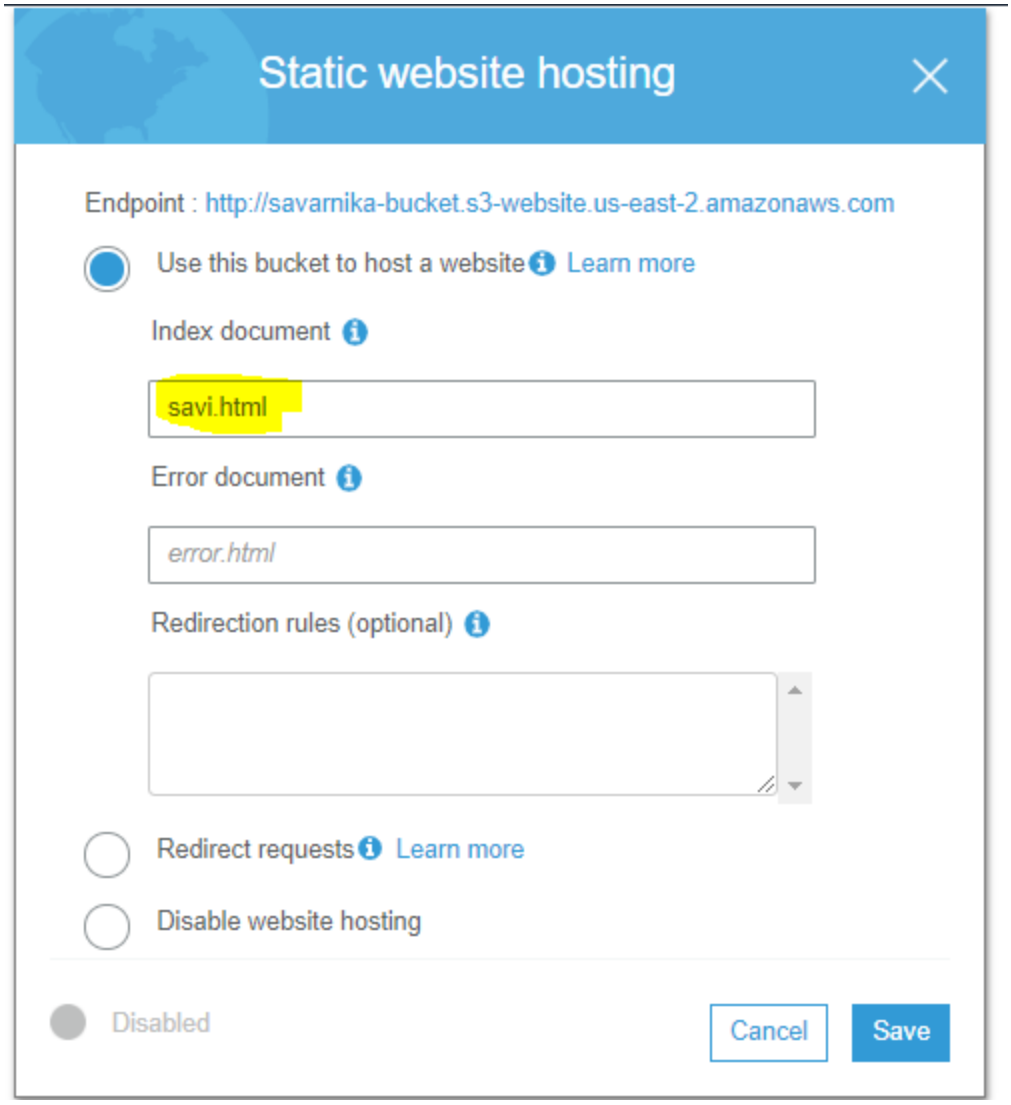

Upload
Select files
2 Set permissions
3 Set properties
4 Review
Manage users
User ID
Objects
Object permissions
seshuchitti2(Owner)
Read
Read
Write
Access for other AWS account
Add account
Account
Objects
Object permissions
Manage public permissions
The block public access settings turned on for this bucket prevent granting public access.
Do not grant public read access to this object(s) (Recommended)
Upload
Previous
Next


Upload
Select files
Set permissions
3 Set properties
4 Review
Storage class
Choose a storage class based on your use case and access requirements. Learn more or see Amazon S3 pricing
Storage class | Designed for | Availability Zones | Min storage duration | Min billable object size | Monitoring and automation fees | Retrieval fees
Standard | Frequently accessed data | ≥ 3 | - | - | - | -
Intelligent-Tiering | Long-lived data with changing or unknown access patterns | ≥ 3 | 30 days | - | Per-object fees apply | -
Standard-IA | Long-lived, infrequently accessed data | ≥ 3 | 30 days | 128KB | - | Per-GB fees apply
One Zone-IA | Long-lived, infrequently accessed, non-critical data | ≥ 1 | 30 days | 128KB | - | Per-GB fees apply
Glacier | Archive data with retrieval times ranging from minutes to hours | ≥ 3 | 90 days | 40KB | - | Per-GB fees apply
Upload
Previous
Next

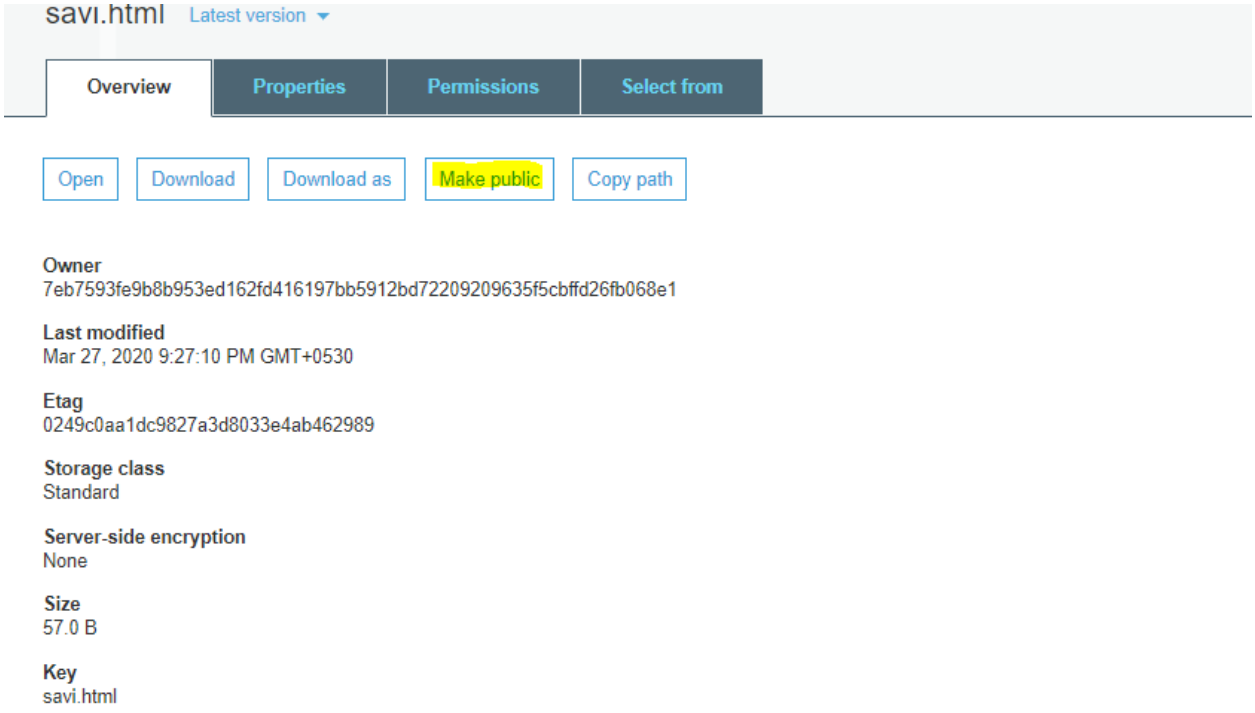

Select files
Set permissions
Set properties
4 Review
Files
Edit
1 Files
Size: 57.0 B
Permissions
Edit
1 grantees
Properties
Edit
Encryption
No
Storage class
Standard
Metadata
Tag
Previous
Upload



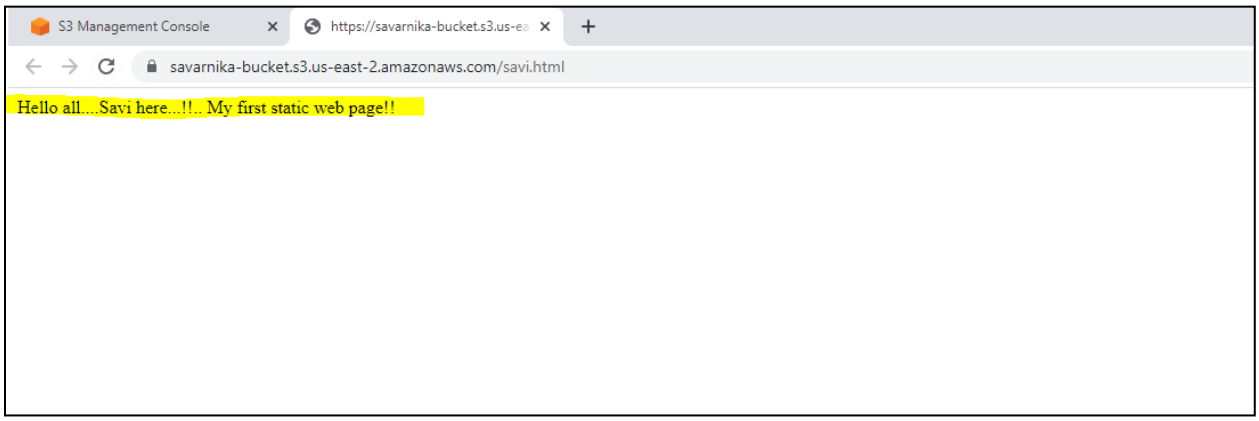

3. Enabling Static Website




4. Making object public

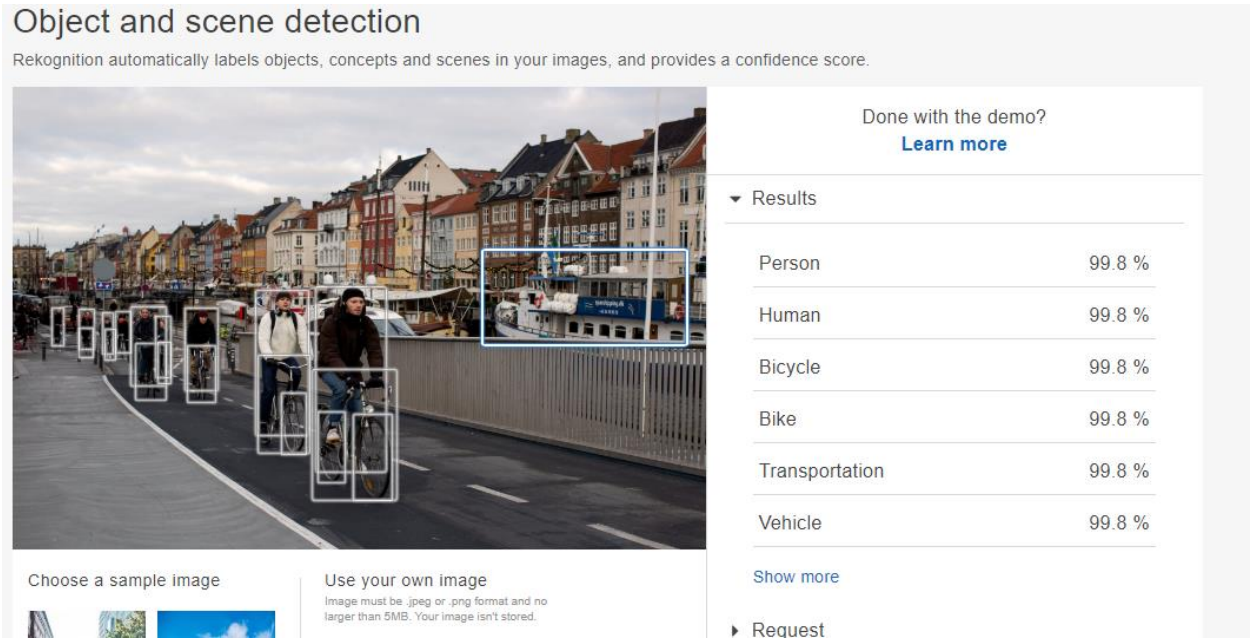



5. Checking the S3 link on the browser

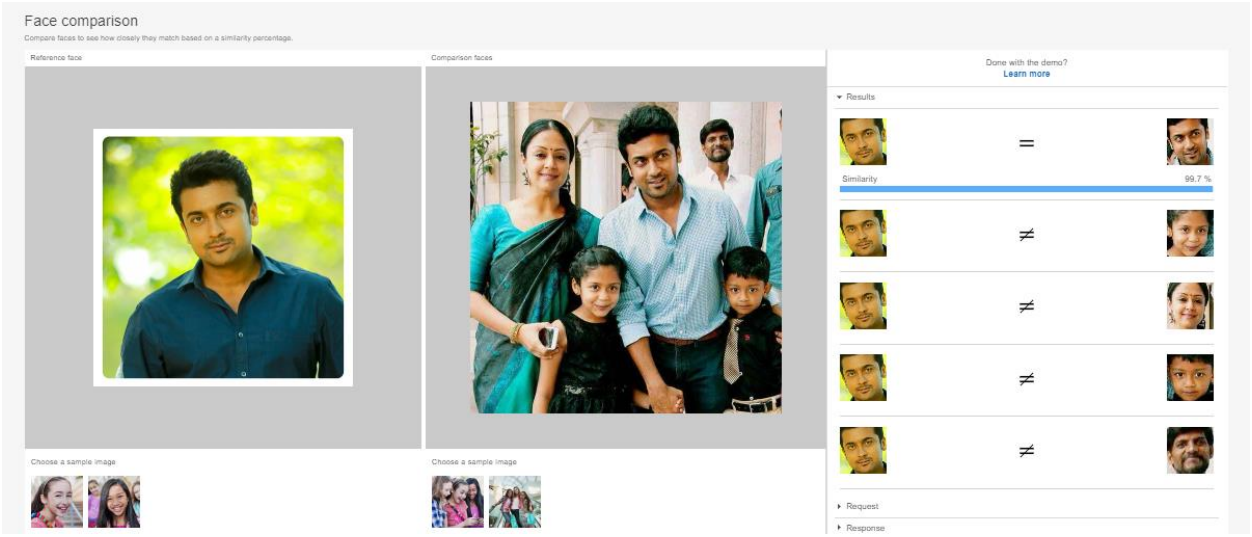



>>Screenshots for Rekognition

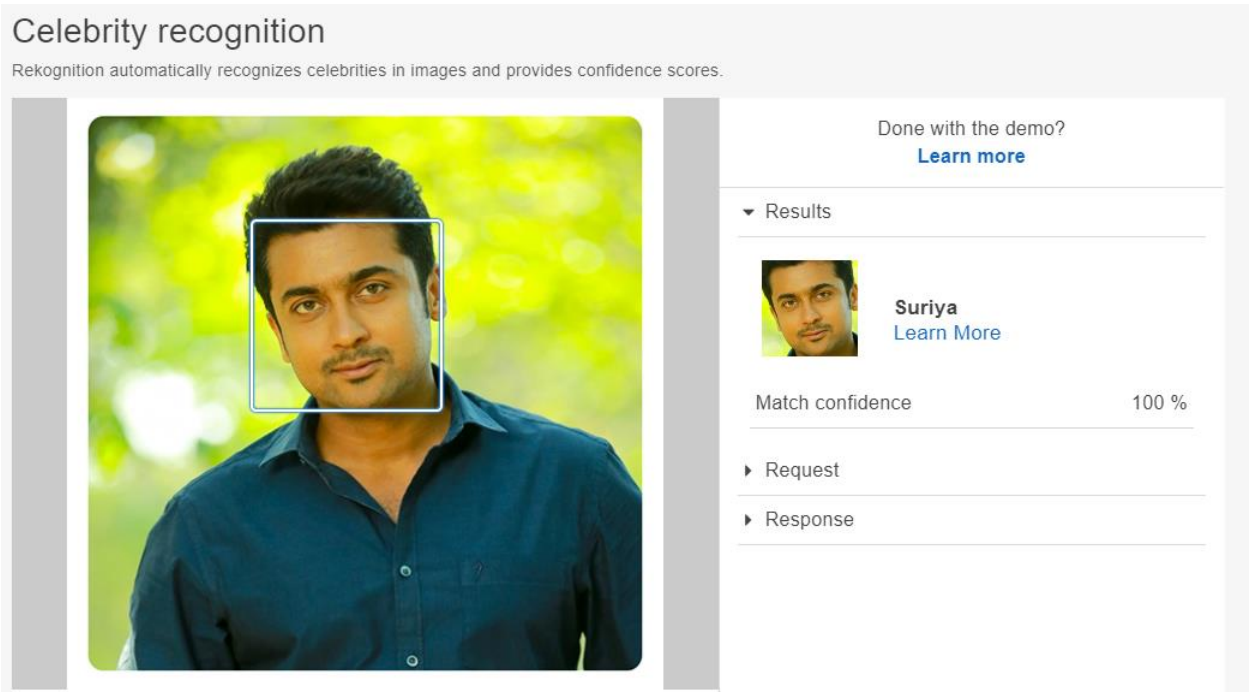
1. Face detect



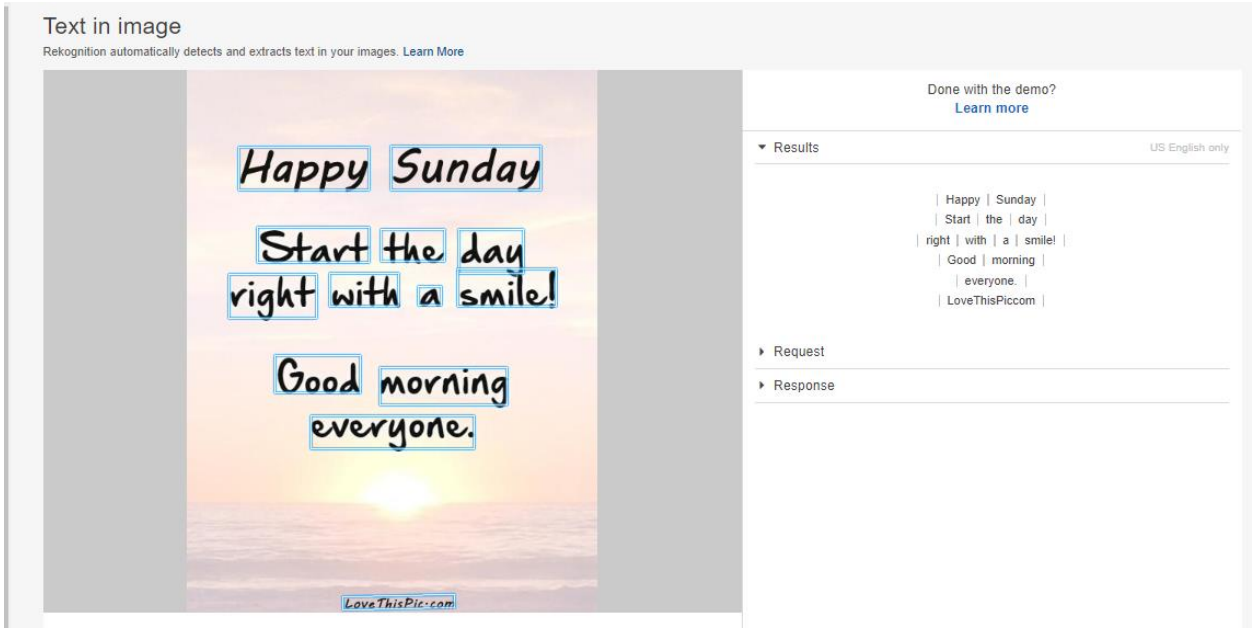

2. Face Compare




3. Celebrity Recognition




4. Text in image




>>Screenshots for EC2 and S3

1. Installing aws-sdk

```
[ec2-user@ip-172-31-0-194 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 hav
e 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 cachi
ng, 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 Im
port/Export)
Package guzzle/guzzle is abandoned, you should avoid using it. Use guzzlehttp/guzzle instead.
Writing lock file
Generating autoload files
```

2. Installing php

```
[ec2-user@ip-172-31-0-194 ~]$ sudo yum install php
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
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: php-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
--> Running transaction check
---> Package php-cli.x86_64 0:5.4.16-46.amzn2.0.2 will be installed
---> Package php-common.x86_64 0:5.4.16-46.amzn2.0.2 will be installed
--> Processing Dependency: libzip.so.2()(64bit) for package: php-common-5.4.16-46.amzn2.0.2.x86_64
--> Running transaction check
---> Package libzip010-compat.x86_64 0:0.10.1-9.amzn2.0.5 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================================================
 Package                    Arch              Version                       Repository              Size
================================================================================================================
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                                                              15 MB/s | 4.7 MB  00:00:00
Running transaction check
```

3. index.php file code

```php
<?php
error_reporting(0);

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

use Aws\S3\S3Client;
use Aws\Rekognition\RekognitionClient;
$bucket = 'savarnika-bucket';
$keyname = 'savi.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;
}
~
~
~
~
~
~
~
~
~
~
~
"index.php" 32L, 717C
```

4. Upload success screenshot

```
[ec2-user@ip-172-31-19-136 face]$ sudo php index.php
Image upload done... Here is the URL: https://savarnika-bucket.s3.us-east-2.amazonaws.com/savi.jpg[ec2-user@ip-172-31-19-136 face]$
```

>>Screenshots for EC2 and Rekognition

1.Faces detect success Screenshot