

# Face-Detection App Project on AWS

SRIDHARAM SRIKANTH

Note :- the-sri-sri is my genuine account name for this project please consider it.

This projects contains several steps including screenshots for the respective steps.

STEP 1 : - Log in to Amazon AWS and dashboard.

STEP 2 : - Creating EC2 instance .

STEP 3 : - Connecting to EC2 instance using PuTTY .

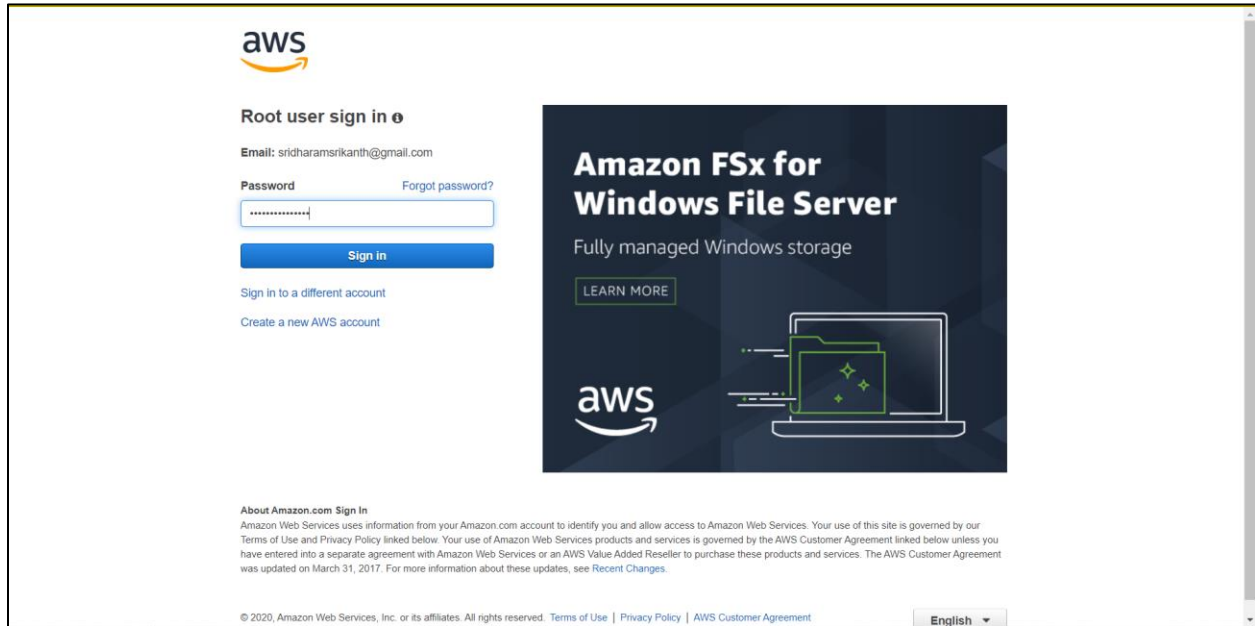
STEP 4 : - Creating S3 bucket.

STEP 5 : - Connecting S3 to EC2 and uploading Objects to S3.

STEP 6 : - Tour to AWS rekognition services .

STEP 7 : - : Using AWS Rekognition from EC2.

**Step 1 : - log in to the Amazon AWS services with your Root user Email and password after activating your Amazon AWS account.**



The screenshot shows the Amazon AWS Root user sign-in page. At the top left is the AWS logo. Below it, the text "Root user sign in" is followed by a small icon. The email field is pre-filled with "Email: sridharamsrikanth@gmail.com". The password field is masked with "\*\*\*\*\*" and has a "Forgot password?" link to its right. A blue "Sign in" button is below the password field. Below the button are two links: "Sign in to a different account" and "Create a new AWS account". To the right of the sign-in form is a promotional banner for "Amazon FSx for Windows File Server" with the text "Fully managed Windows storage" and a "LEARN MORE" button. The banner also features the AWS logo and an illustration of a laptop with a folder icon. At the bottom left, there is a section titled "About Amazon.com Sign in" with a paragraph of text and links to "Terms of Use", "Privacy Policy", and "AWS Customer Agreement". At the bottom right, there is a language selector set to "English".

aws

Root user sign in

Email: sridharamsrikanth@gmail.com

Password [Forgot password?](#)

\*\*\*\*\*

Sign in

[Sign in to a different account](#)

[Create a new AWS account](#)

**Amazon FSx for Windows File Server**

Fully managed Windows storage

[LEARN MORE](#)

aws

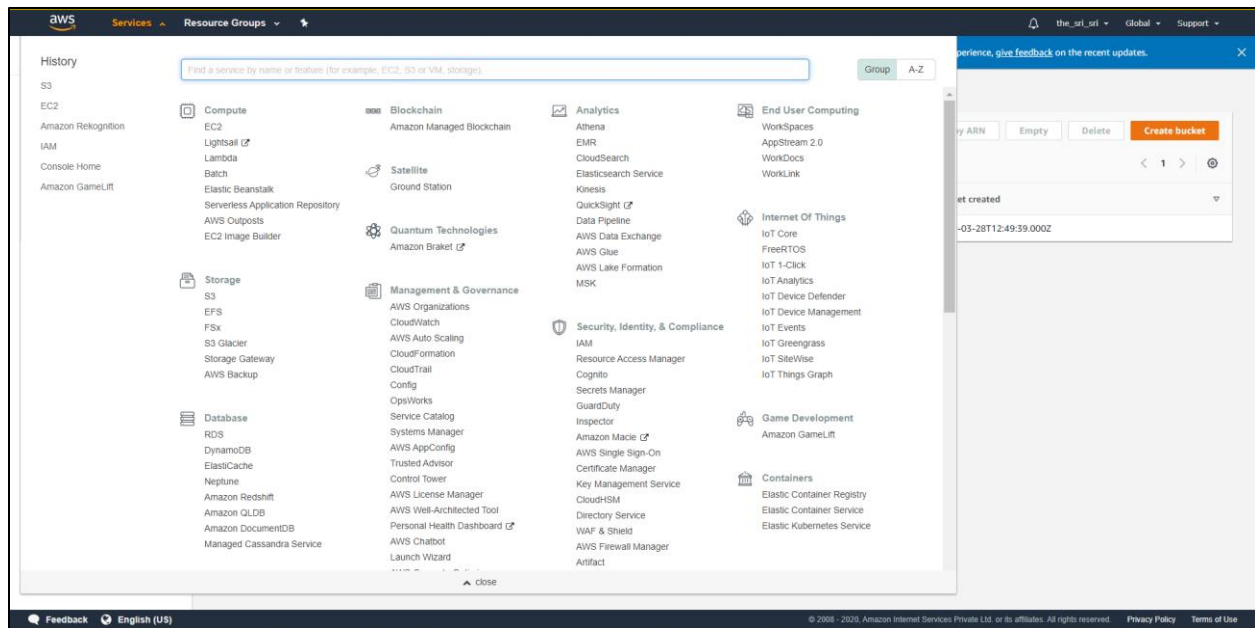
**About Amazon.com Sign in**

Amazon Web Services uses information from your Amazon.com account to identify you and allow access to Amazon Web Services. Your use of this site is governed by our [Terms of Use](#) and [Privacy Policy](#) linked below. Your use of Amazon Web Services products and services is governed by the [AWS Customer Agreement](#) linked below unless you have entered into a separate agreement with Amazon Web Services or an AWS Value Added Reseller to purchase these products and services. The AWS Customer Agreement was updated on March 31, 2017. For more information about these updates, see [Recent Changes](#).

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Terms of Use](#) | [Privacy Policy](#) | [AWS Customer Agreement](#)

English

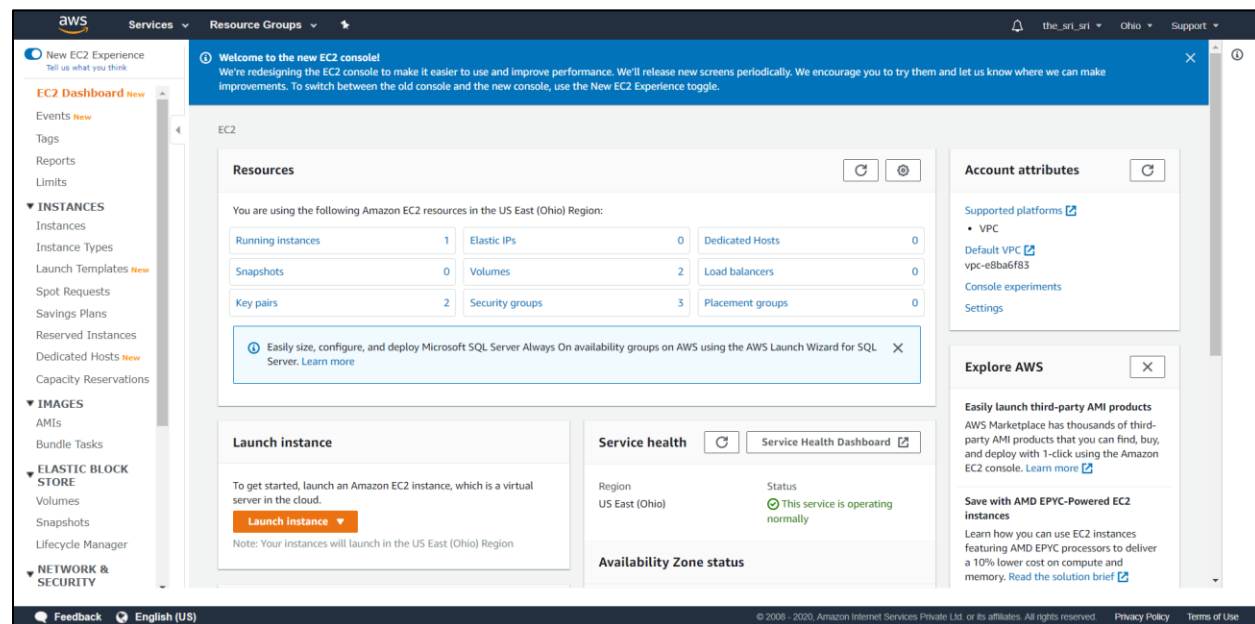
After logging in the dashboard of the services provided by Amazon AWS will be presented as below partitioned by their service .



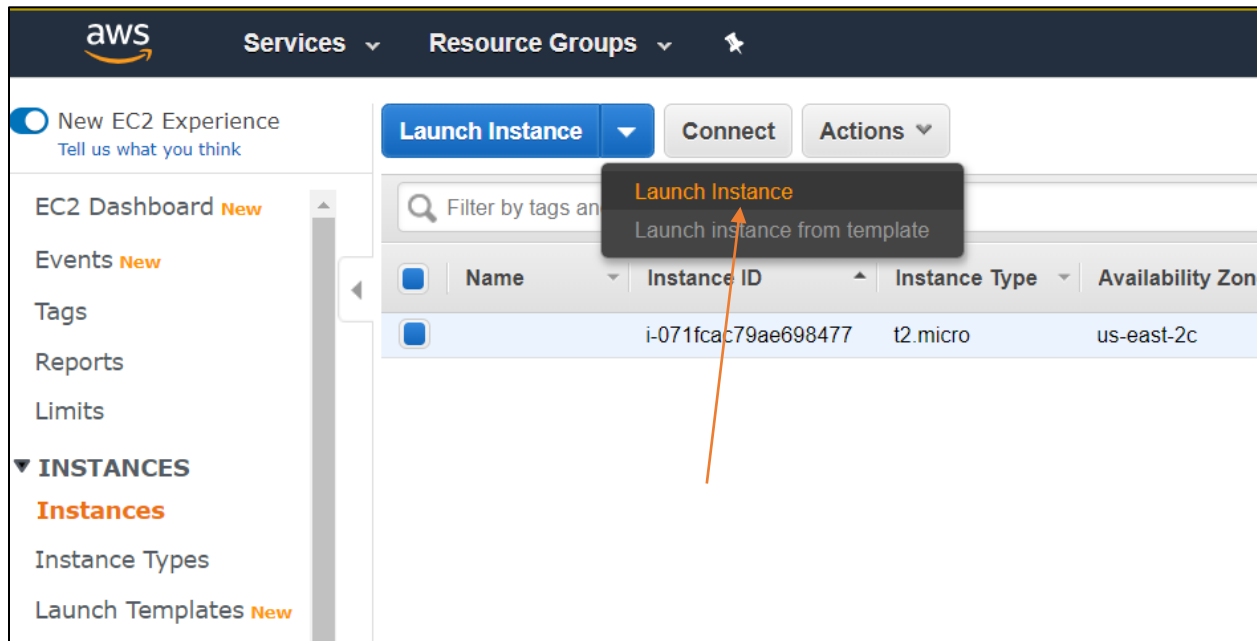
**STEP 2 :** - create an EC2 instance . follow the steps under given to create an EC2 instance .

Note: - here second instance has been created for demonstration purpose .

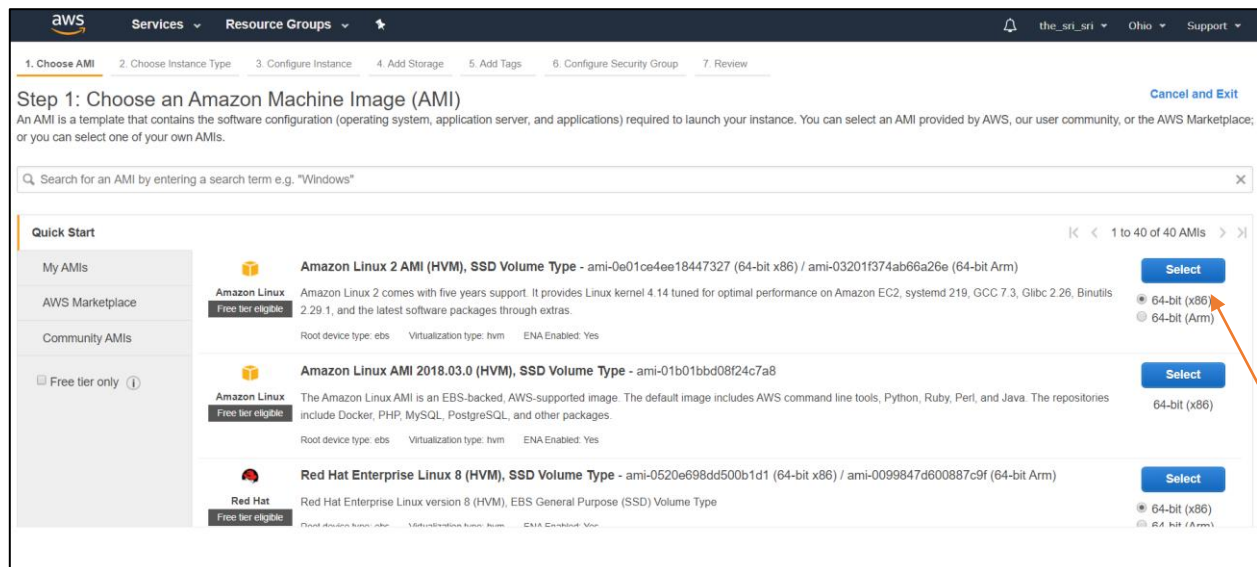
## EC2 dashboard



- select Launch instance under launch instance button .



- Select an Operating System to work upon (here we have selected **Amazon Linux 2 Ami** )



- Select required Instance type i.e. Type & No. of CPU , Memory needed based on your project needs. (for this project we have selected **t2.micro** )  
Now click **Next** .

**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

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Configure Instance Details](#)

- Now Configure your Subnets in this step according to your needs (we **haven't** done any changes as per our project) click **next**.

**Step 3: Configure Instance Details**

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of Instances:  [Launch into Auto Scaling Group](#)

Purchasing option: ☐ Request Spot instances

Network:  [Create new VPC](#)

Subnet:  [Create new subnet](#)

Auto-assign Public IP:

Placement group: ☐ Add instance to placement group

Capacity Reservation:  [Create new Capacity Reservation](#)

IAM role:  [Create new IAM role](#)

Shutdown behavior:

Stop - Hibernate behavior: ☐ Enable hibernation as an additional stop behavior

Enable termination protection: ☐ Protect against accidental termination

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Configure Subnets](#)

- Add Storage Details i.e. size of storage you need for your project (By default its 8GiB we **haven't done** any changes ) click **Next**.

**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 GiB 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](#)

- here we **haven't added** any tags for our EC2 instance just click **NEXT**.

**Step 5: Add Tags**  
A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. A copy of a tag can be applied to volumes, instances or both. Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key (128 characters maximum)	Value (256 characters maximum)	Instances	Volumes
This resource currently has no tags			

Choose the [Add tag](#) button or [click to add a Name tag](#).  
Make sure your [IAM policy](#) includes permissions to create tags.

[Add Tag](#) (Up to 50 tags maximum)

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Configure Security Group](#)

- Configure your security group according to the Type of EC2 we have created .(we have created **SSH type i.e. Linux based OS** and for that **port No. is 22**) click **PREVIEW & LAUNCH**.

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.

[Cancel](#)
[Previous](#)
[Review and Launch](#)

- In this step **preview** all your **changes and configurations** for confirmation and then click **launch**.

aws Services Resource Groups the\_sri\_sri Ohio Support

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

### Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

**Improve your instances' security. Your security group, launch-wizard-2, is open to the world.**

Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only. You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. [Edit security groups](#)

[Edit AMI](#)

**AMI Details**

**Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0e01ce4ee18447327**

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: ebs Virtualization type: hvm

[Edit instance type](#)

**Instance Type**

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

[Edit security groups](#)

**Security Groups**

[Cancel](#)
[Previous](#)
[Launch](#)

- On clicking **LAUNCH** a dialogue box will appear for generating private Key for the instance. Select **Create a new key Pair** option from the **drop down** and provide a **name** of your choice for your key . then **download key pair** and select **launch instance**.

**Select an existing key pair or create a new key pair** ✕

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Create a new key pair ▼

**Key pair name**  
aws-demo

Download Key Pair

**You have to download the private key file (\*.pem file) before you can continue. Store it in a secure and accessible location. You will not be able to download the file again after it's created.**

Cancel Launch Instances

- the **Final Launch status** for your **EC2 instance** will be like **this** .

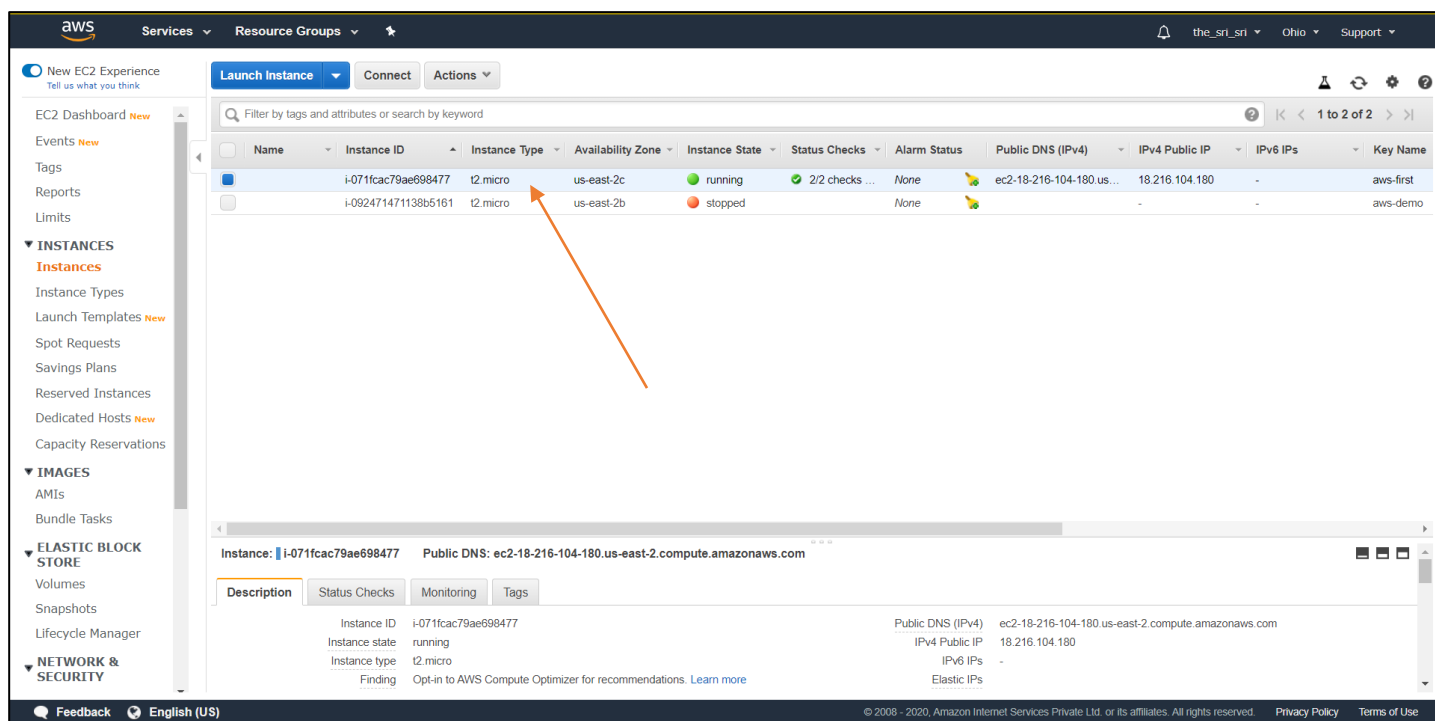
**Launch Status**

✓ **Your instances are now launching**  
The following instance launches have been initiated: [i-092471471138b5161](#) [View launch log](#)

**Get notified of estimated charges**  
[Create billing alerts](#) to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).

- The instances you create will be displayed on the dashboard of EC2 .





### STEP 3 : - Connecting to EC2 instance using PuTTY.

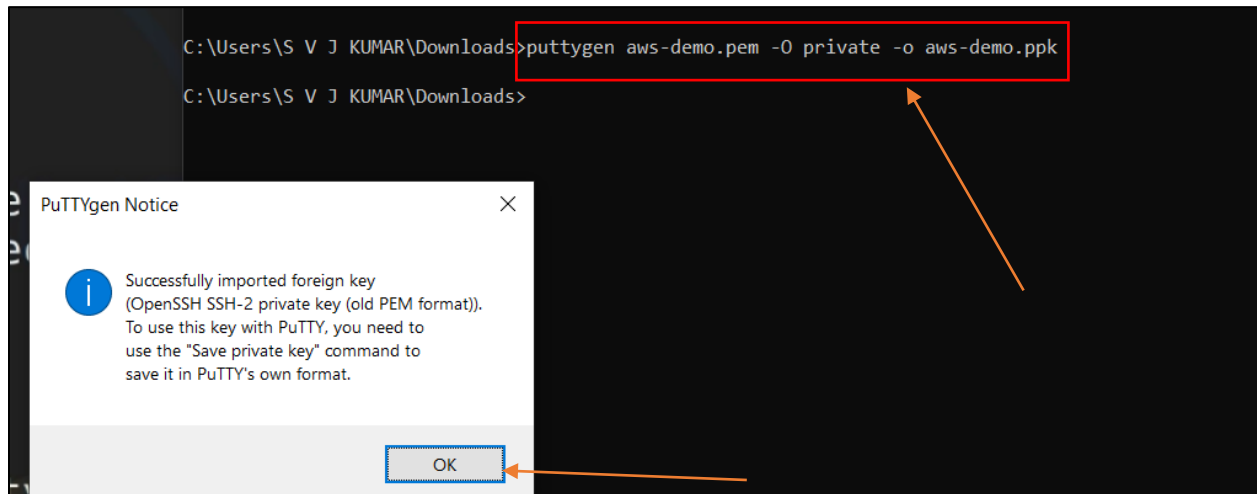
Now that we have created EC2 instance we should be connecting to this Instance using **PuTTY**.

#### Why PuTTY?

We have created our instance on Linux based OS, to communicate with that Linux based Instance we need an external application s\w i.e. PuTTY .

PuTTY is available for all the OS (MAC OS, WIN, Linux etc. ) you should be downloading the software according to your base OS of the system you are using to connect to the Instance based on Linux OS.

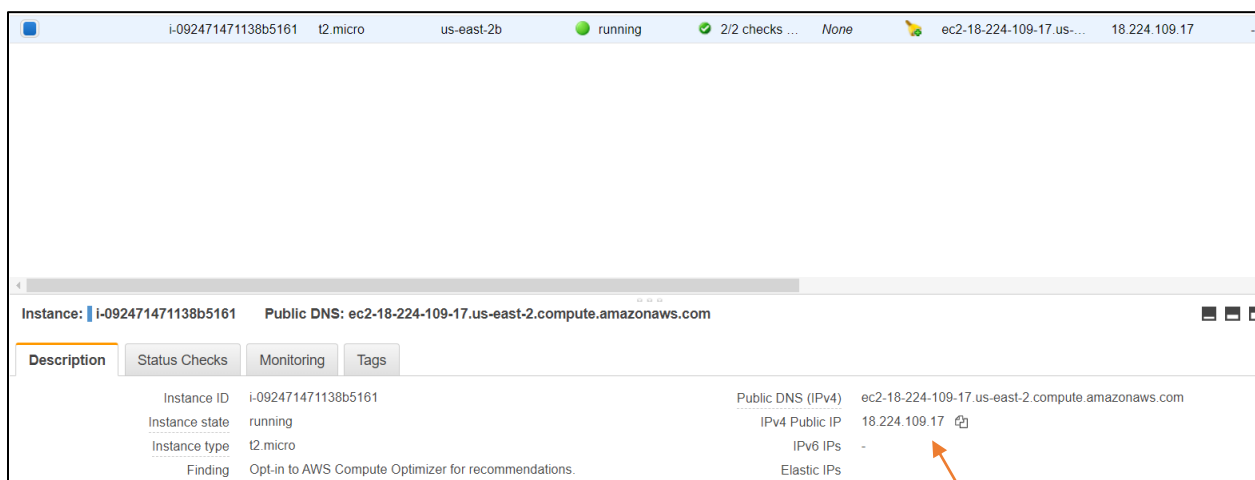
In the process of launching our instance we have downloaded a **Key file** which is in .pem format and for this key file to be used with PuTTY we need to convert it to .ppk format, we will be doing this using **keygen** command in our windows **CMD prompt** .



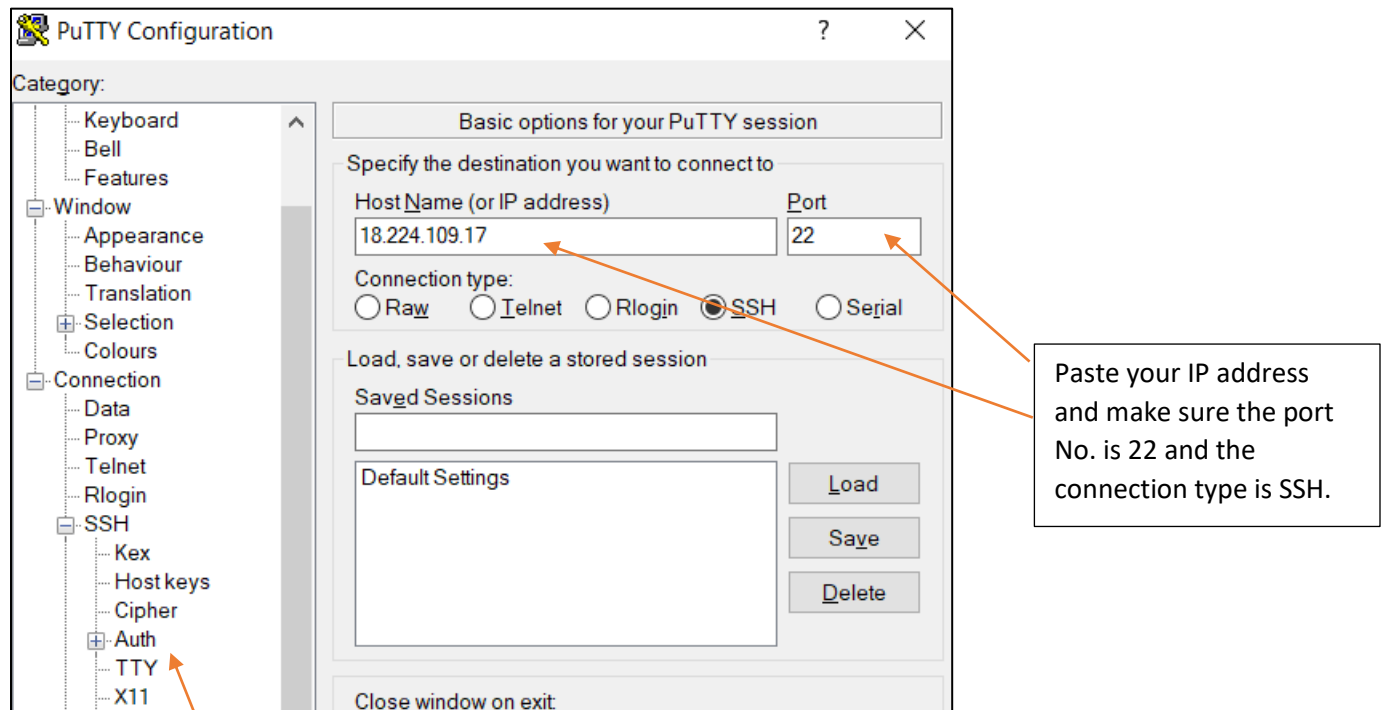
Type the **highlighted command** in the cmd prompt and click ok to get .ppk format of your **key file** .

After downloading and installing the PuTTY launch it .

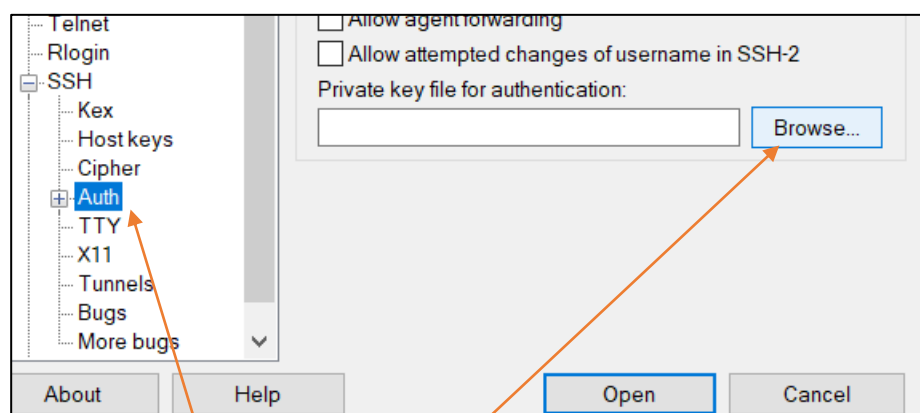
Copy the public IP address of your instance from the EC2 instance dashboard.



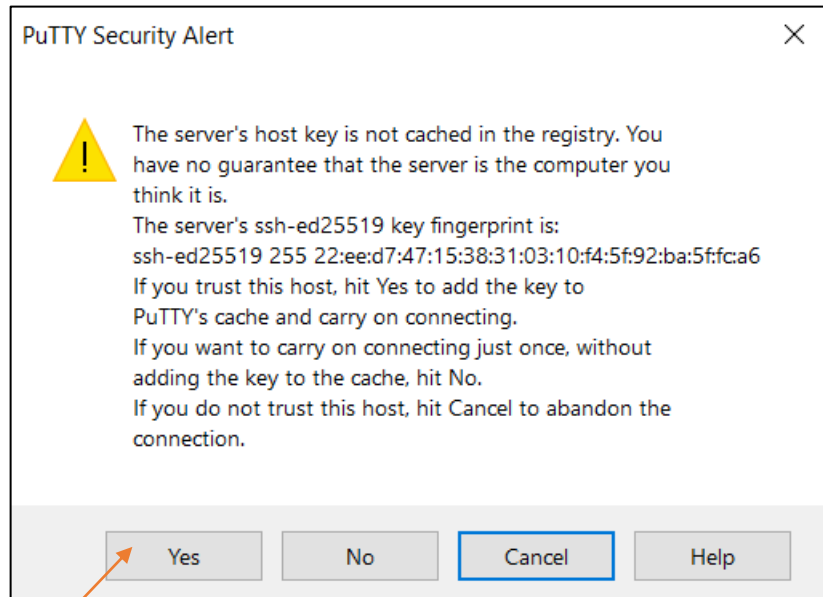
And **paste it** on the **PuTTY Configuration GUI**.



Now select  
Connection/SSH/auth  
From the path tree on  
the left side



Under Auth Browse open your **.ppk file of private key** file and click Open.

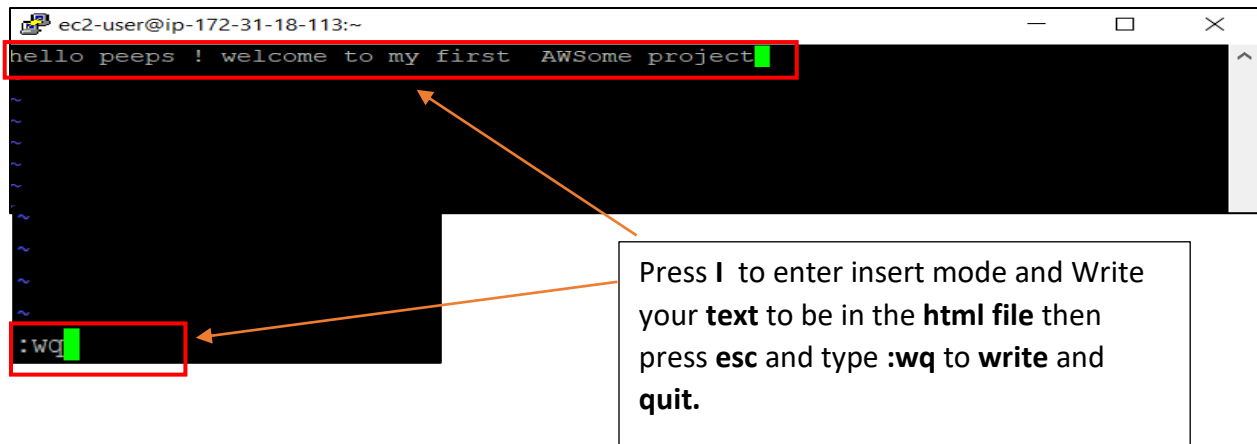


Select **yes** for this security alert dialogue box. And your will be seeing an **Command Line Interface of PuTTY** .

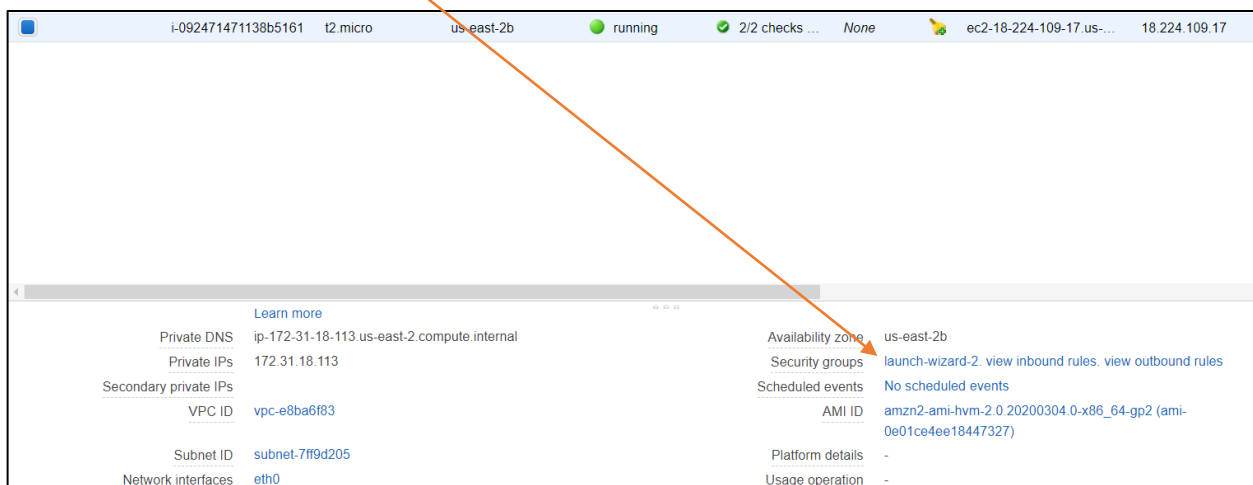
Type **the sequence of commands** in the command line interface of **PuTTY** to connect to **EC2 instance** .

```
ec2-user@ip-172-31-18-113:~  
login as: ec2-user  
Authenticating with public key "imported-openssh-key"  
  
  _ | _ | _ )  
  _ | ( _ - /   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-172-31-18-113 ~]$ sudo yum install httpd
```

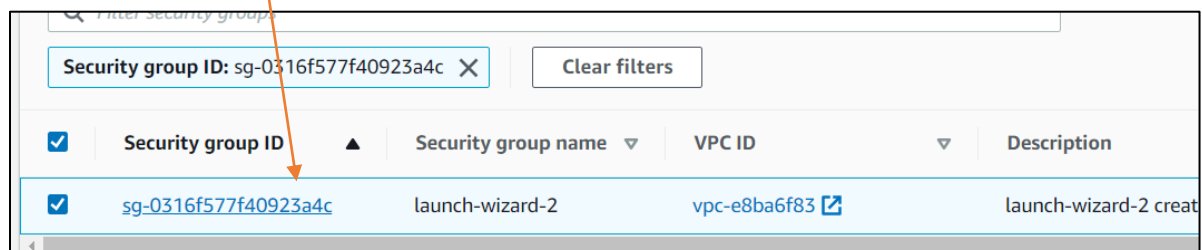
```
ec2-user@ip-172-31-18-113:~  
[ec2-user@ip-172-31-18-113 ~]$ sudo service httpd start  
Redirecting to /bin/systemctl start httpd.service  
[ec2-user@ip-172-31-18-113 ~]$ sudo service httpd status  
Redirecting to /bin/systemctl status httpd.service  
● httpd.service - The Apache HTTP Server  
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor prese  
t: disabled)  
   Active: active (running) since Fri 2020-03-27 16:30:55 UTC; 14s ago  
     Docs: man:httpd.service(8)  
  Main PID: 3578 (httpd)  
    Status: "Total requests: 0; Idle/Busy workers 100/0; Requests/sec: 0; Bytes se  
rved/sec:  0 B/sec"  
    CGroup: /system.slice/httpd.service  
            └─3578 /usr/sbin/httpd -DFOREGROUND  
              └─3579 /usr/sbin/httpd -DFOREGROUND  
                └─3580 /usr/sbin/httpd -DFOREGROUND  
                  └─3581 /usr/sbin/httpd -DFOREGROUND  
                    └─3582 /usr/sbin/httpd -DFOREGROUND  
                      └─3583 /usr/sbin/httpd -DFOREGROUND  
  
Mar 27 16:30:54 ip-172-31-18-113.us-east-2.compute.internal systemd[1]: Start...  
Mar 27 16:30:55 ip-172-31-18-113.us-east-2.compute.internal systemd[1]: Start...  
Hint: Some lines were ellipsized, use -l to show in full.  
[ec2-user@ip-172-31-18-113 ~]$ sudo vim /var/www/html/index.html
```



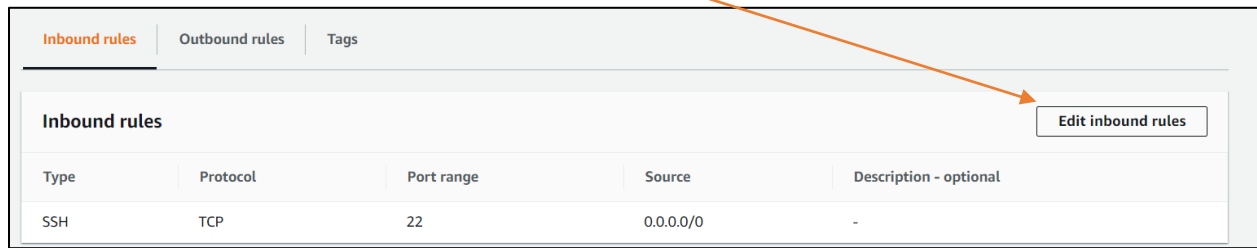
Now go back to the **EC2 instance dashboard** and **select your instance** before selecting **launch-wizards-2** from the **security groups** tag.



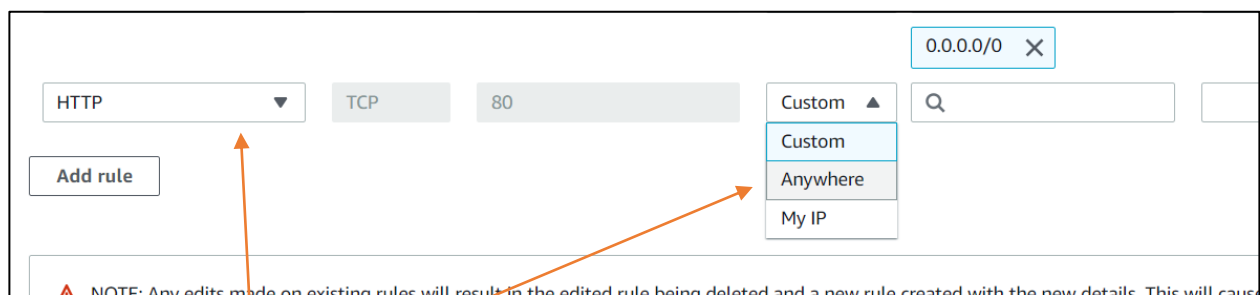
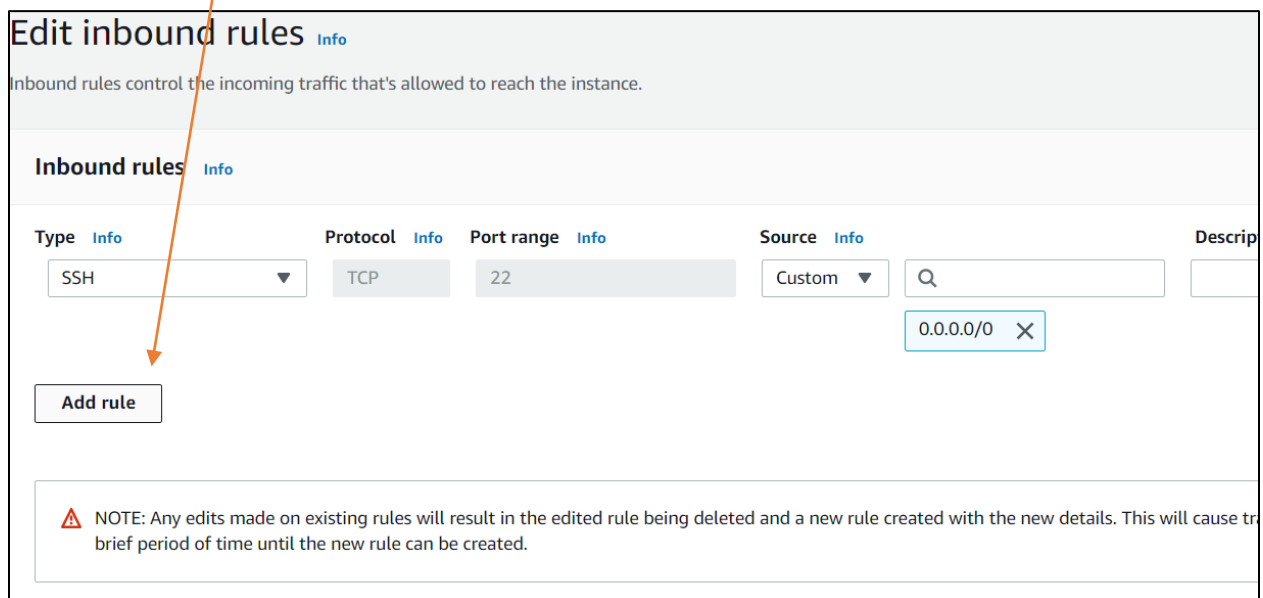
now select **security group id** of your instance from the given list of ids



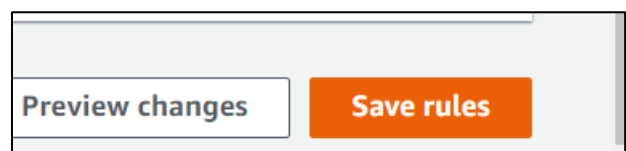
now select **Edit inbound rules** .



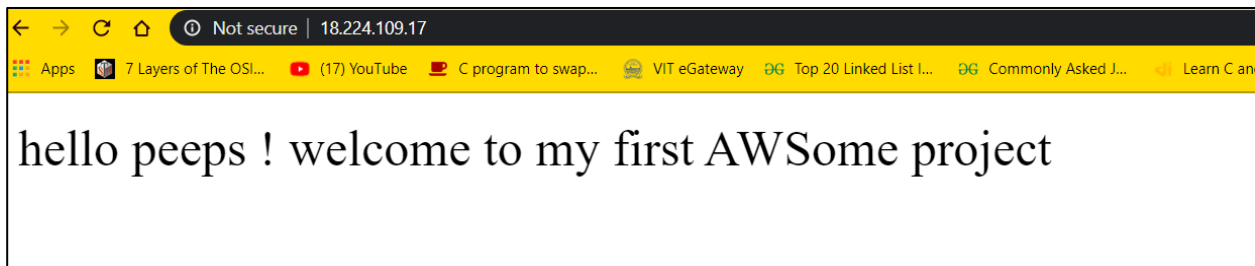
now press **Add Rules** button



and select **type** as **HTML** and source as **anywhere** for the **new rule** to make your **html webpage accessible across the globe** .



now search for your **IP address** in your **web browser** and watch your static **web page** on the internet **live**.

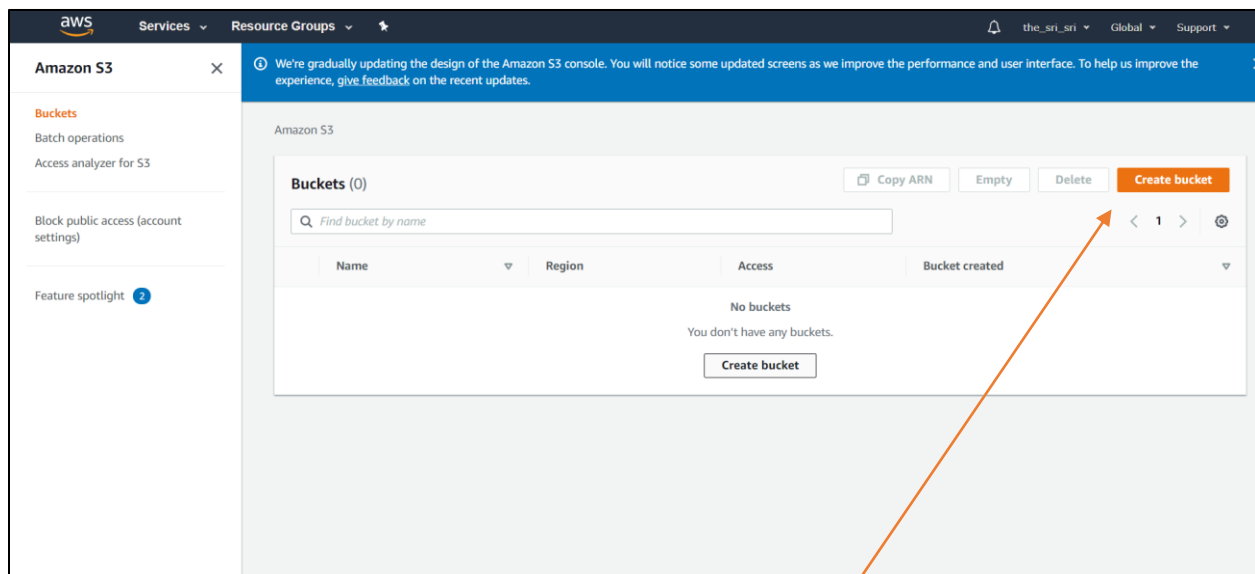


#### STEP 4 : - Creating S3 bucket.

Buckets in S3 are the collection of objects (every file i.e. .jpg , .pdf,.html, .docx etc. is treated as an object in S3) with unique IDs.

Buckets do not have any region it agrees to be global unlike the objects in the bucket which are accessible from specified region only .

The under given is the **Dashboard** of **AWS S3**.



To create your first bucket start your steps by clicking the **“Create Bucket”** button on the right corner of the **Dashboard**.



**Bucket's name** is **Unique** in nature, as it is shown globally , to recognize It in uniquely one must provide a unique name to the bucket. Then select **region** for your bucket because (The user interface shows all your buckets, in all regions. But buckets exist in a specific region and you need to specify that region when you create a bucket.) and select '**create bucket**'.

**General configuration**

Bucket name  
unique-bucket-name

Bucket name must be unique and must not contain spaces or uppercase letters. [See rules for bucket naming](#)

Region  
US East (Ohio) us-east-2

**Bucket settings for Block Public Access**

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

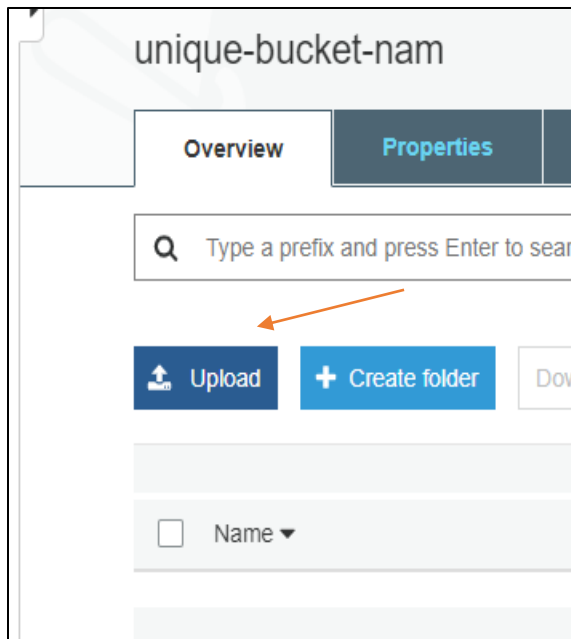
☒ **Block all public access**  
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- ☒ **Block public access to buckets and objects granted through *new* access control lists (ACLs)**  
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- ☒ **Block public access to buckets and objects granted through *any* access control lists (ACLs)**  
S3 will ignore all ACLs that grant public access to buckets and objects.
- ☒ **Block public access to buckets and objects granted through *new* public bucket or access point policies**  
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- ☒ **Block public and cross-account access to buckets and objects through *any* public bucket or access point policies**  
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

► **Advanced settings**

Cancel Create bucket

To upload your files into bucket click the **“upload”** button on the right side of the **GUI**



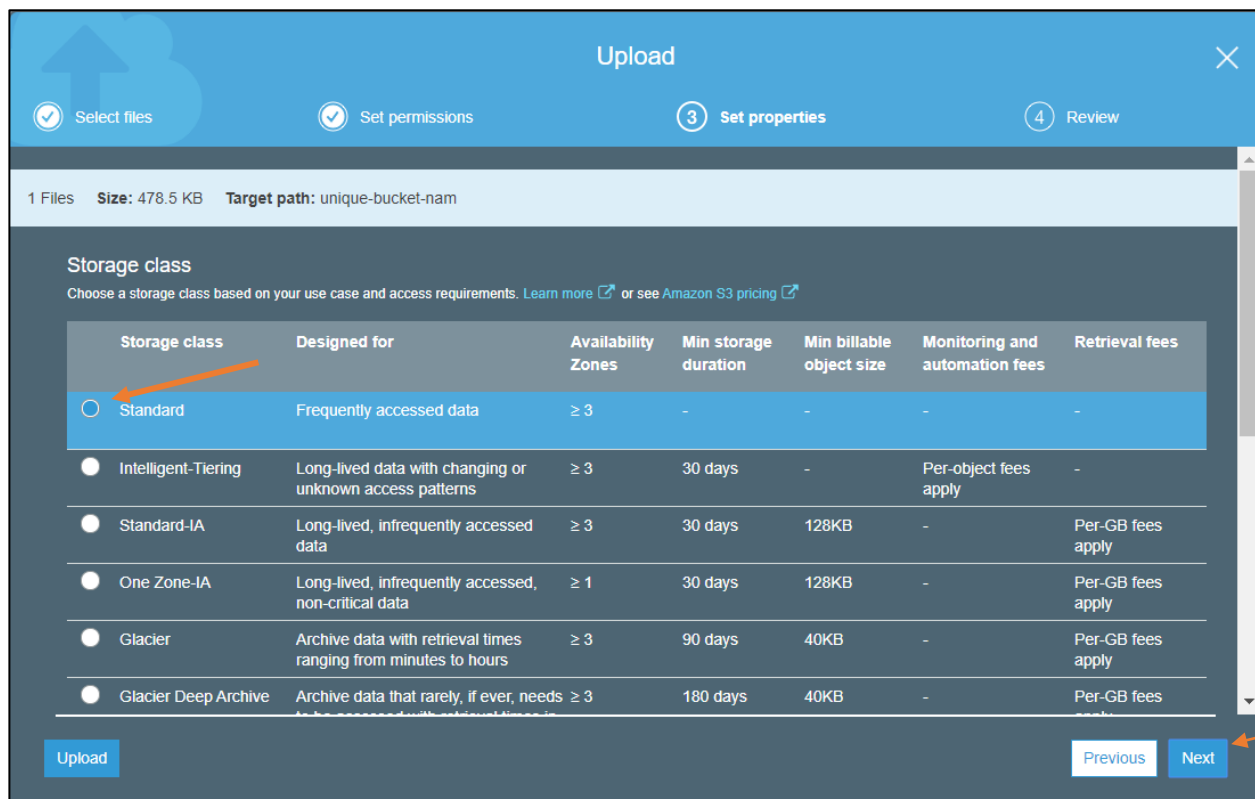
Or you can create **folder** to make **collections** of your **objects**.

Then drag drop or select your **object** using file selector from the local disk and then press **next**.

The on the next step toggle **access permission** of your file as per your need and click **“Next”**.

The **“Properties”** lets you select the type of **storage class** you want for your data/ **file**.

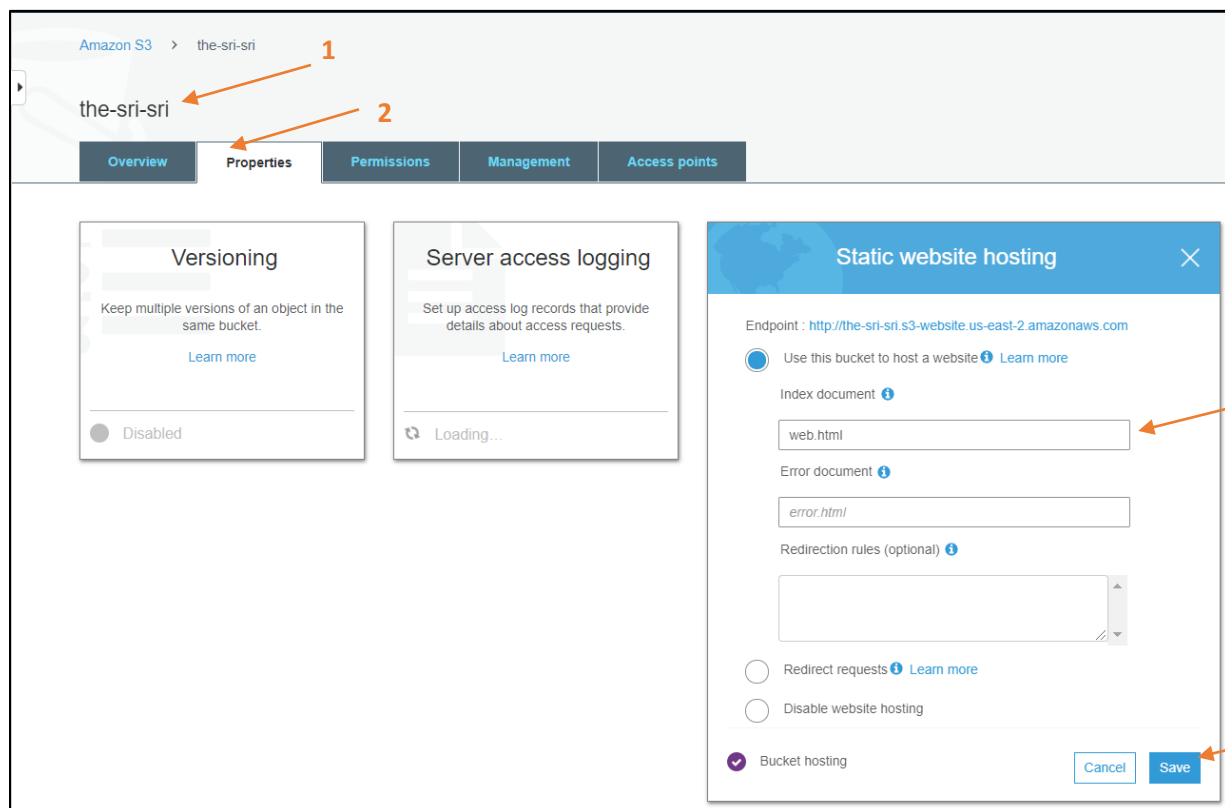
1. Standard : Data **accessed frequently** are stored in all the **Availability zones** of the **region** you selected earlier.

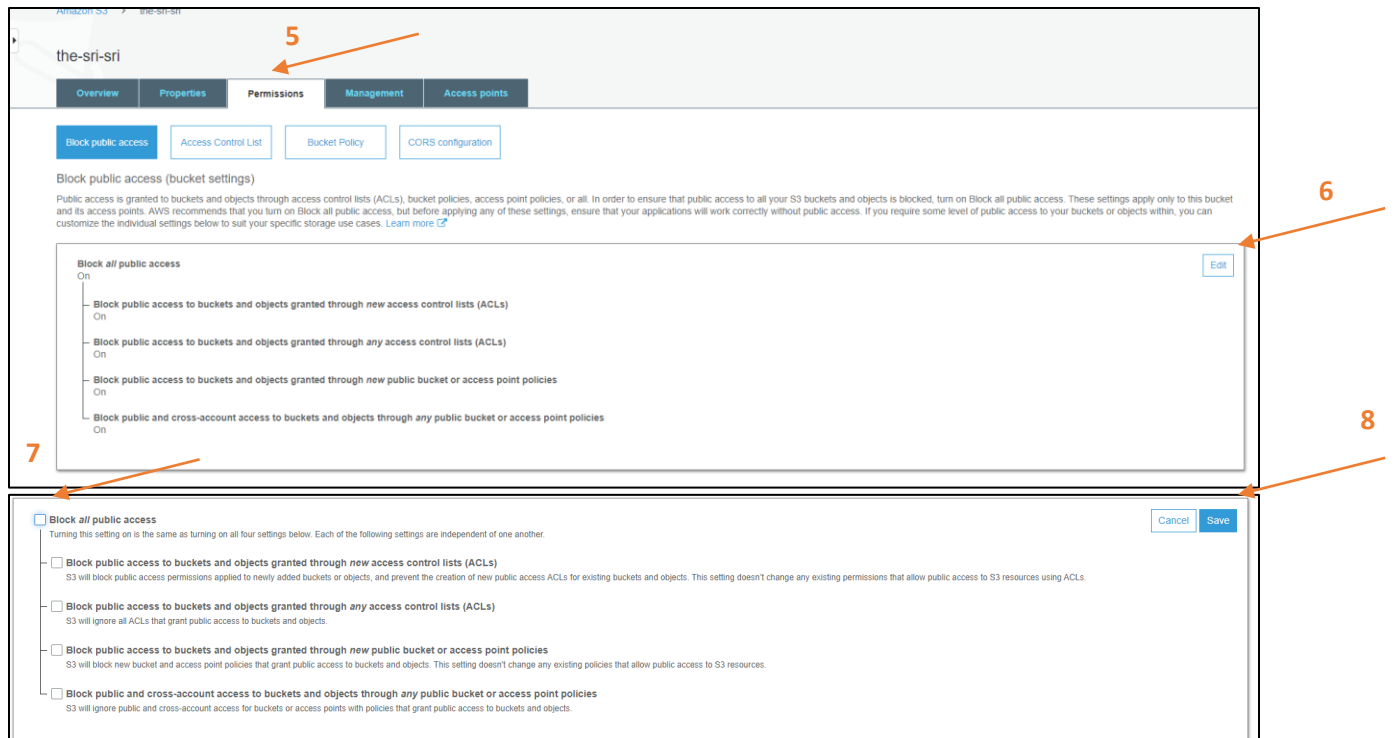


2. Intelligent-tiering : for the data to be accessed with unknown patterns . but critical / Important.
3. Standard-IA : data to be stored for a long time , infrequently accessed , and non-critical.
4. Glacier : archived data , very rarely accessed , takes more time for retrieval.
5. Glaciers Deep Archive : archived data takes more time for retrieval but kept for 2X days than Glaciers class.

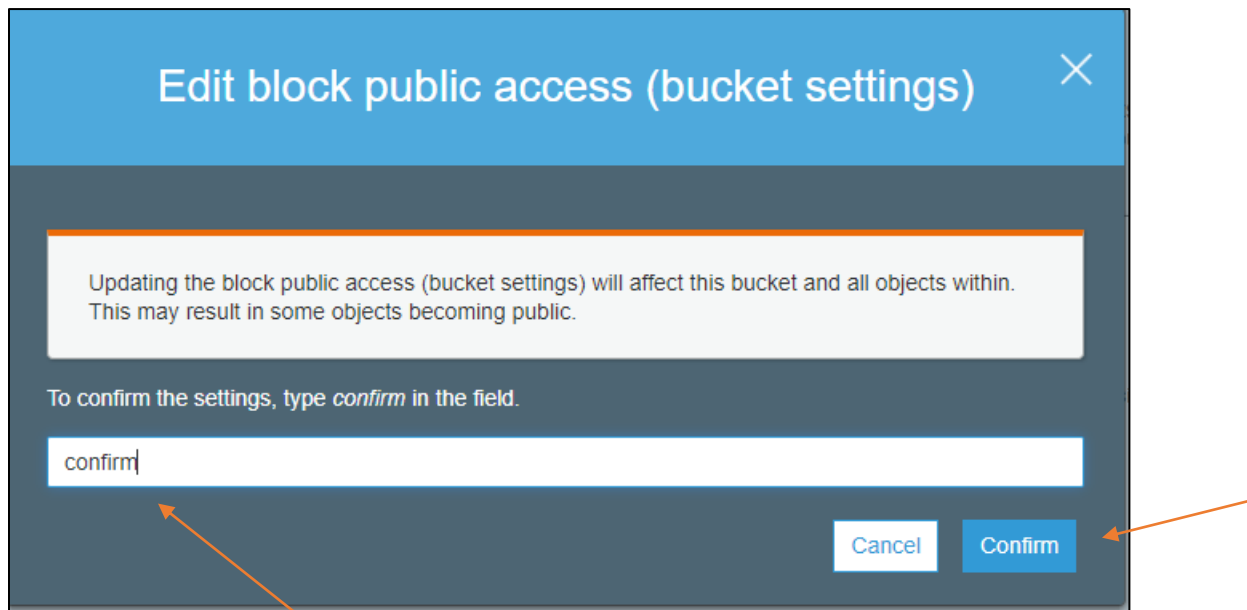
**Follow under given steps to Host your file to the web.**

1. Select your bucket .
2. Select Properties tab.
3. Select **Static web hosting** box. Name your file
4. and click **save** .
5. select **Permissions** tab.
6. and select the **edit** option
7. and toggle **on** to **off**.
8. Click **save**

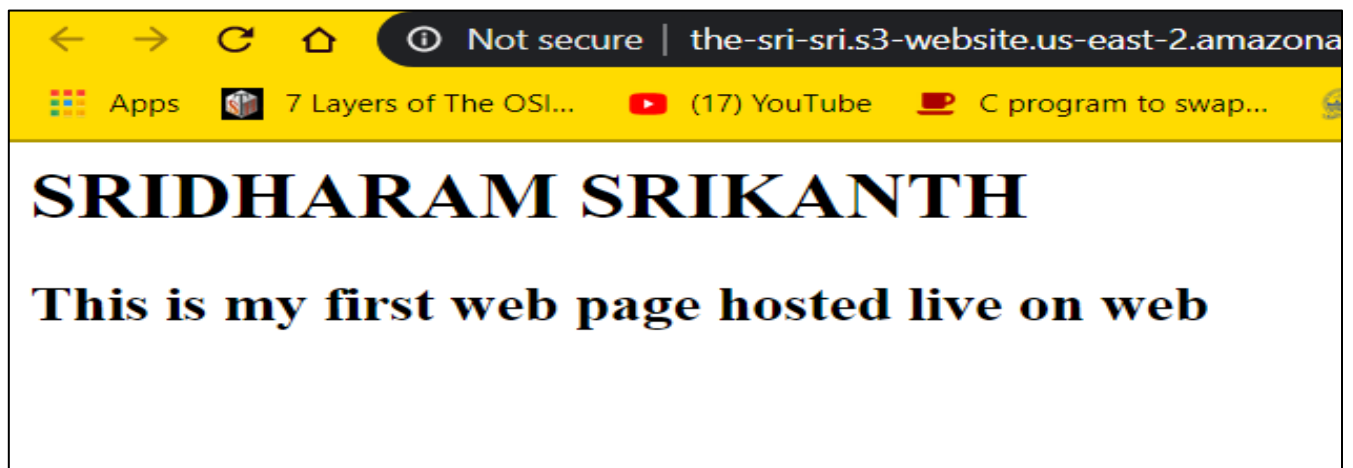
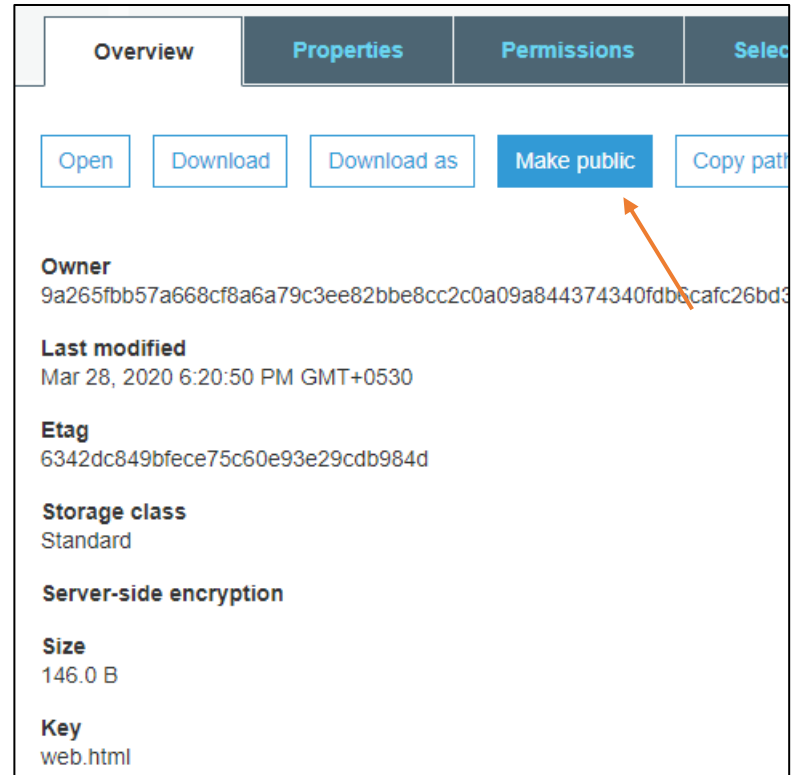
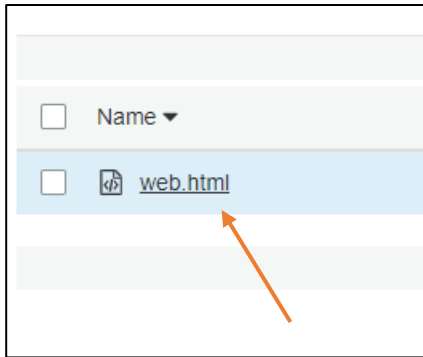




9. Type **“Confirm”** in the dialogue box appeared and press **confirm**.



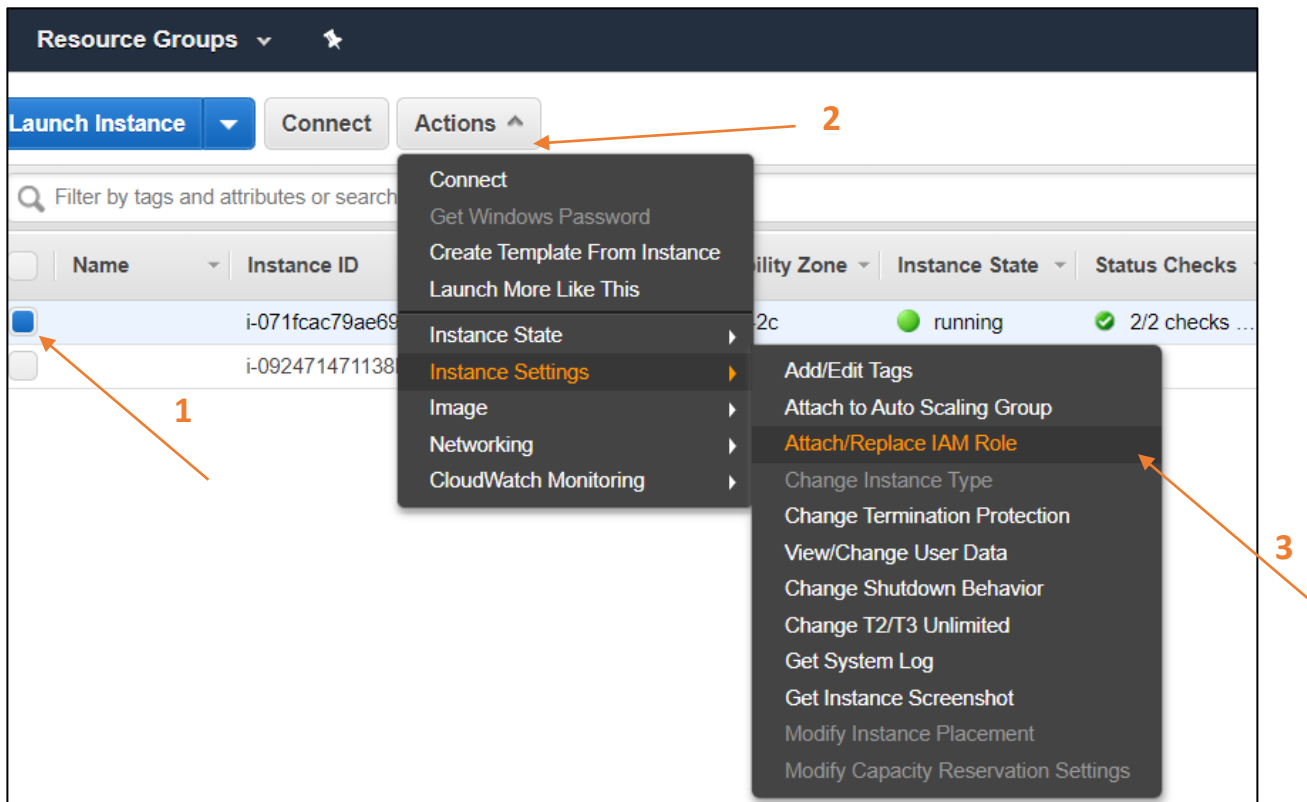
10. Select the **file** you want to make public and the click on **“Make Public”** tab.



## STEP 5 : - Connecting S3 to EC2 and uploading Objects to S3

### Create IAM Role .

Select your EC2 instance and then **follow bellow illustrated steps.**



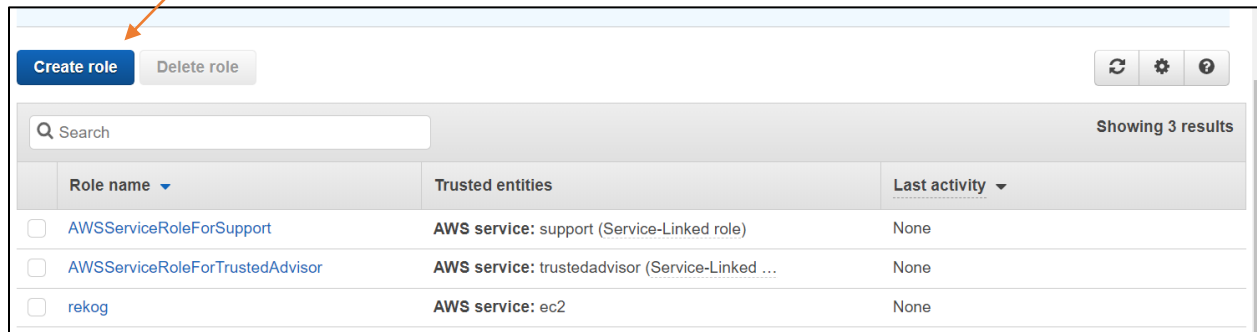
If you don't have any IAM roles, choose Create new IAM role to create a role in the IAM console.  
The IAM role you choose will replace the existing role.




i-071fcac79ae698477 () ⓘ

[Create new IAM role](#) ⓘ

4

5





<a href="#">Create role</a> <a href="#">Delete role</a>		  	
<input type="text" value="Search"/>		Showing 3 results	
Role name	Trusted entities	Last activity	
<input type="checkbox"/> <a href="#">AWSServiceRoleForSupport</a>	AWS service: support (Service-Linked role)	None	
<input type="checkbox"/> <a href="#">AWSServiceRoleForTrustedAdvisor</a>	AWS service: trustedadvisor (Service-Linked ...)	None	
<input type="checkbox"/> <a href="#">rekog</a>	AWS service: ec2	None	


## Create role


1 2 3 4

### Select type of trusted entity

**AWS service**  
EC2, Lambda and others

**Another AWS account**  
Belonging to you or 3rd party

**Web identity**  
Cognito or any OpenID provider

**SAML 2.0 federation**  
Your corporate directory

Allows AWS services to perform actions on your behalf. [Learn more](#)

### Choose a use case

#### Common use cases

##### EC2

Allows EC2 instances to call AWS services on your behalf.

##### Lambda

Allows Lambda functions to call AWS services on your behalf.

#### Or select a service to view its use cases

<a href="#">API Gateway</a>	<a href="#">CodeDeploy</a>	<a href="#">EMR</a>	<a href="#">KMS</a>	<a href="#">RoboMaker</a>
<a href="#">AWS Backup</a>	<a href="#">CodeGuru</a>	<a href="#">ElastiCache</a>	<a href="#">Kinesis</a>	<a href="#">S3</a>
<a href="#">AWS Chatbot</a>	<a href="#">CodeStar Notifications</a>	<a href="#">Elastic Beanstalk</a>	<a href="#">Lambda</a>	<a href="#">SMS</a>
<a href="#">AWS CloudFormation</a>	<a href="#">CodeBuild</a>	<a href="#">Elastic Container Service</a>	<a href="#">Amazon Rekognition</a>	<a href="#">AWS IoT</a>

\* Required

[Cancel](#)

[Next: Permissions](#)

8

## Create role

1234

▼ Attach permissions policies

Choose one or more policies to attach to your new role.

Create policy

↺

Filter policies ▼

Q s3

Showing 4 results

	Policy name ▼	Used as
<input type="checkbox"/>	▶  AmazonDMSRedshiftS3Role	None
<input type="checkbox"/>	▶  AmazonS3FullAccess	Permissions policy (1)
<input type="checkbox"/>	▶  AmazonS3ReadOnlyAccess	None
<input type="checkbox"/>	▶  QuickSightAccessForS3StorageManagementAnalyticsReadOnly	None

\* Required

CancelPreviousNext: Tags

## Tags (optional)

are key-value pairs you can add to your role. Tags can include user information, such as an email address, or can be descriptive, such as a job. You can use the tags to organize, track, or control access for this role. [Learn more](#)

Value (optional)

Remove

new key

add 50 more tags.

CancelPreviousNext: Review



Create role

view

Provide the required information below and review this role before you create it.

**Role name\***  13

Use alphanumeric and '+,=,.,@,-,\_' characters. Maximum 64 characters.

**Role description**

Maximum 1000 characters. Use alphanumeric and '+,=,.,@,-,\_' characters.

**Trusted entities** AWS service: ec2.amazonaws.com

**Policies** Policies not attached

**Permissions boundary** Permissions boundary is not set

Tags were added.

Required Cancel Previous Create role 14

Go to **Attach / Replace IAM Role** page after selecting your **EC2 instance**.

15 Instances > Attach/Replace IAM Role

Attach/Replace IAM Role

Select an IAM role to attach to your instance. If you don't have any IAM roles, choose Create new IAM role to create a role in the IAM console.  
If an IAM role is already attached to your instance, the IAM role you choose will replace the existing role.

Instance ID i-071fcac79ae698477 ⓘ

**IAM role\***  16 Create new IAM role ⓘ

Filter by attributes

**Profile Name**

No Role

rekog

17 Cancel Apply

Open PuTTY and connect to EC2 and follow these below given steps to down a file from web and upload it to S3 through EC2 instance.

```
ec2-user@ip-172-31-44-120:~  
login as: ec2-user  
Authenticating with public key "imported-openssh-key"  
Last login: Fri Mar 27 18:14:51 2020 from 27.62.203.49  
  
  _ | _ | _ )  
  _ | ( _ - /  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-172-31-44-120 ~]$ sudo yum install httpd  
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd  
Package httpd-2.4.41-1.amzn2.0.1.x86_64 already installed and latest version  
Nothing to do  
[ec2-user@ip-172-31-44-120 ~]$ 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  
    php-common.x86_64 0:5.4.16-46.amzn2.0.2  
  
Complete!  
[ec2-user@ip-172-31-44-120 ~]$ curl -sS https://getcomposer.org/installer | php  
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-172-31-44-120 ~]$  
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-172-31-44-120 ~]$ cd /var/www/html  
[ec2-user@ip-172-31-44-120 html]$ cd face  
-bash: cd: face: No such file or directory  
[ec2-user@ip-172-31-44-120 html]$ sudo mkdir face  
[ec2-user@ip-172-31-44-120 html]$ cd face  
[ec2-user@ip-172-31-44-120 face]$ pwd  
/var/www/html/face  
[ec2-user@ip-172-31-44-120 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): Downloading (100%)  
   proc_open(): fork failed - Cannot allocate memory  
   The archive may contain identical file names with different capitalization (which fails on case insensitive filesystems)
```

```
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] [--] [<packages>]...
```

```
[ec2-user@ip-172-31-44-120 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, 13.3766 s, 80.3 MB/s
```

```
[ec2-user@ip-172-31-44-120 face]$ sudo /sbin/mkswap /var/swap.1
```

```
mkswap: /var/swap.1: insecure permissions 0644, 0600 suggested.
```

```
Setting up swapspace version 1, size = 1024 MiB (1073737728 bytes)
```

```
no label, UUID=2052b033-ad5c-4e08-bf13-6c1aa794d8c2
```

```
[ec2-user@ip-172-31-44-120 face]$ sudo /sbin/swapon /var/swap.1
```

```
swapon: /var/swap.1: insecure permissions 0644, 0600 suggested.
```

```
[ec2-user@ip-172-31-44-120 face]$
```

Sudo wget {link of image to be downloaded}

```
Try 'wget --help' for more options.
```

```
[ec2-user@ip-172-31-44-120 ~]$ sudo wget https://i.pinimg.com/originals/b9/7e/a3/b97ea33b5842c7894b804923c6c05580.jpg
```

```
--2020-03-28 15:46:17-- 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 0.05s
```

```
2020-03-28 15:46:17 (4.54 MB/s) - 'b97ea33b5842c7894b804923c6c05580.jpg' saved [215551/215551]
```

```
[ec2-user@ip-172-31-44-120 ~]$ ls
```

```
b97ea33b5842c7894b804923c6c05580.jpg composer.phar
```

```
[ec2-user@ip-172-31-44-120 ~]$ sudo m^C
```

```
[ec2-user@ip-172-31-44-120 ~]$ sudo mv b97ea33b5842c7894b804923c6c05580.jpg s.jpg
```

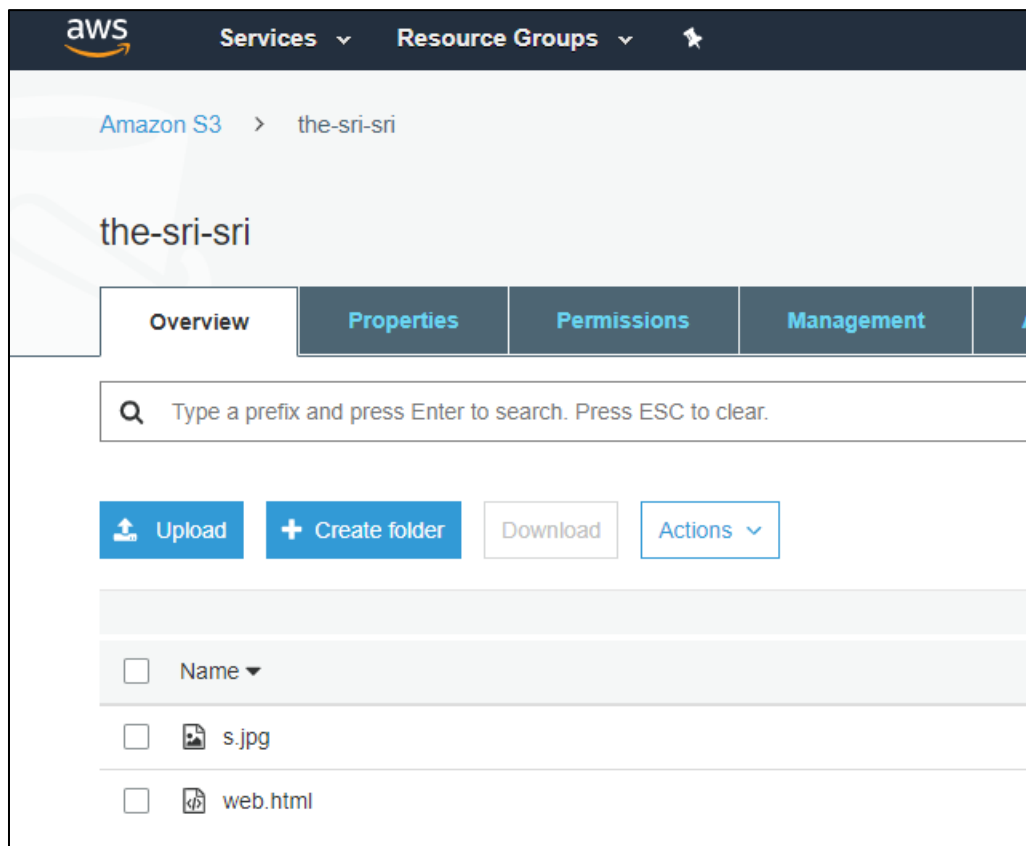
```
[ec2-user@ip-172-31-44-120 ~]$ ls
```

```
composer.phar s.jpg
```

```
[ec2-user@ip-172-31-44-120 ~]$
```

```
ec2-user@ip-172-31-44-120:/var/www/html/face
Redirecting to /bin/systemctl start httpd.service
[ec2-user@ip-172-31-44-120 ~]$ cd /var/www/html
[ec2-user@ip-172-31-44-120 html]$ cd face
[ec2-user@ip-172-31-44-120 face]$ sudo php -d memory_limit=-1 ~/composer.phar re
quire aws/aws-sdk-php
Using version ^2.8 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/gu
zzle instead.
Generating autoload files
[ec2-user@ip-172-31-44-120 face]$ ls
o97ea33b5842c7894b804923c6c05580.jpg.1  composer.lock  s.jpg  vendor
composer.json  index.php  team.jpg
[ec2-user@ip-172-31-44-120 face]$ sudo php index.php
Image upload done... Here is the URL: https://the-sri-sri.s3.us-east-2.amazonaws
.com/s.jpg[ec2-user@ip-172-31-44-120 face]$
```

your file has been uploaded to S3 check in your S3 bucket



## STEP 6: - A tour to AWS rekognition.

### ✓ Object and scene detection

The screenshot shows the AWS Rekognition console interface. On the left is a navigation menu with options like Custom Labels, Demos, and Object and scene detection (which is highlighted). The main content area is titled 'Object and scene detection' and includes a description: 'Rekognition automatically labels objects, concepts and scenes in your images, and provides a confidence score.' Below this is a large image of a city street with several cars and a person jumping over a low wall. Blue bounding boxes are drawn around the cars and the person. To the right of the image is a 'Results' section with a table of detected objects and their confidence scores.

Object	Confidence Score
Car	98.8 %
Automobile	98.8 %
Transportation	98.8 %
Vehicle	98.8 %
Human	98.3 %
Person	98.3 %

Below the main image, there are two options: 'Choose a sample image' with a small thumbnail, and 'Use your own image' with an 'Upload' button and a note that the image must be .jpg or .png format and no larger than 5MB. At the bottom right, there are links for 'Request' and 'Response'.

### ✓ Facial Analysis

The screenshot shows the AWS Rekognition console interface for the 'Facial analysis' demo. The left navigation menu highlights 'Facial analysis'. The main content area is titled 'Facial analysis' and includes a description: 'Get a complete analysis of facial attributes, including confidence scores.' Below this is a large image of a woman wearing sunglasses and driving a yellow car. A blue bounding box is drawn around her face. To the right of the image is a 'Results' section with a table of detected facial attributes and their confidence scores.

Attribute	Confidence Score
looks like a face	99.9 %
appears to be female	99.9 %
age range	17 - 29 years old
smiling	91.7 %
appears to be happy	99.5 %
wearing glasses	99.8 %

Below the main image, there are two options: 'Choose a sample image' with a small thumbnail, and 'Use your own image' with an 'Upload' button and a note that the image must be .jpg or .png format and no larger than 5MB. At the bottom right, there are links for 'Request' and 'Response'.

## ✓ Celebrity Recognition

The screenshot shows the Amazon Rekognition Celebrity Recognition demo. The interface includes a sidebar with navigation links for Amazon Rekognition services, a main content area with a large image of Jeff Bezos, and a results panel on the right. The results panel shows a match for Jeff Bezos with a 100% confidence score.

**Amazon Rekognition**

Services ▾ Resource Groups ▾

the\_srl\_srl ▾ Ohio ▾ Support ▾

Amazon Rekognition

Custom Labels <sup>New</sup>

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


Feedback English (US)

**Celebrity recognition**

Rekognition automatically recognizes celebrities in images and provides confidence scores.

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

▼ Results

 **Jeff Bezos**  
[Learn More](#)

Match confidence 100 %

► Request

► Response

Choose a sample image

Use your own image  
Images must be .png or .jpg format and no larger than 5MB. Your image isn't shared.

[Upload](#) or drag and drop

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

## ✓ Face comparison

The screenshot shows the Amazon Rekognition Face Comparison demo. The interface includes a sidebar with navigation links for Amazon Rekognition services, a main content area with two images of a young girl, and a results panel on the right. The results panel shows a similarity score of 99.8% between the two images.

**Amazon Rekognition**

Services ▾ Resource Groups ▾

the\_srl\_srl ▾ Ohio ▾ Support ▾

Amazon Rekognition

Custom Labels <sup>New</sup>

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

Feedback English (US)

**Face comparison**



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



Reference face



Comparison faces

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

▼ Results

 =   
Similarity 99.8 %

 ≠ 

 ≠ 

► Request

► Response

Choose a sample image

Choose a sample image

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



## ✓ Text in image

The screenshot shows the AWS Rekognition 'Text in image' demo page. The interface includes a sidebar with navigation links for Amazon Rekognition, Custom Labels, Demos, and various services like Object and scene detection, Image moderation, Facial analysis, etc. The main content area displays a sample image of a red mug with a smiley face and the text 'IT'S MONDAY but keep Smiling'. The text is highlighted with bounding boxes. To the right, the 'Results' section shows the detected text: 'IT'S', 'MONDAY', 'but', 'keep', and 'Smiling'. Below the sample image, there are options to 'Choose a sample image' or 'Use your own image' with an 'Upload' button.

## STEP 7: Using AWS Rekognition .

input Image :-



## Output :-

```
ec2-user@ip-172-31-44-120:/var/www/html/face
[ec2-user@ip-172-31-44-120 face]$ sudo service httpd start
Redirecting to /bin/systemctl start httpd.service
[ec2-user@ip-172-31-44-120 face]$ cd /var/www/html/face
[ec2-user@ip-172-31-44-120 face]$ ls
Anupama-Premam.jpg      index.php      s.jpg      vendor
b97ea33b5842c7894b804923c6c05580.jpg.1  Japantable.jpg team.jpg
composer.json          lastindex.php  tele.php
composer.lock          newindex.php   text.php
[ec2-user@ip-172-31-44-120 face]$ sudo vim newindex.php
[ec2-user@ip-172-31-44-120 face]$ sudo php newindex.php
Image upload done... Here is the URL: https://the-sri-sri.s3.us-east-2.amazonaws
.com/s.jpgTotally there are 9 faces[ec2-user@ip-172-31-44-120 face]$
```