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 .

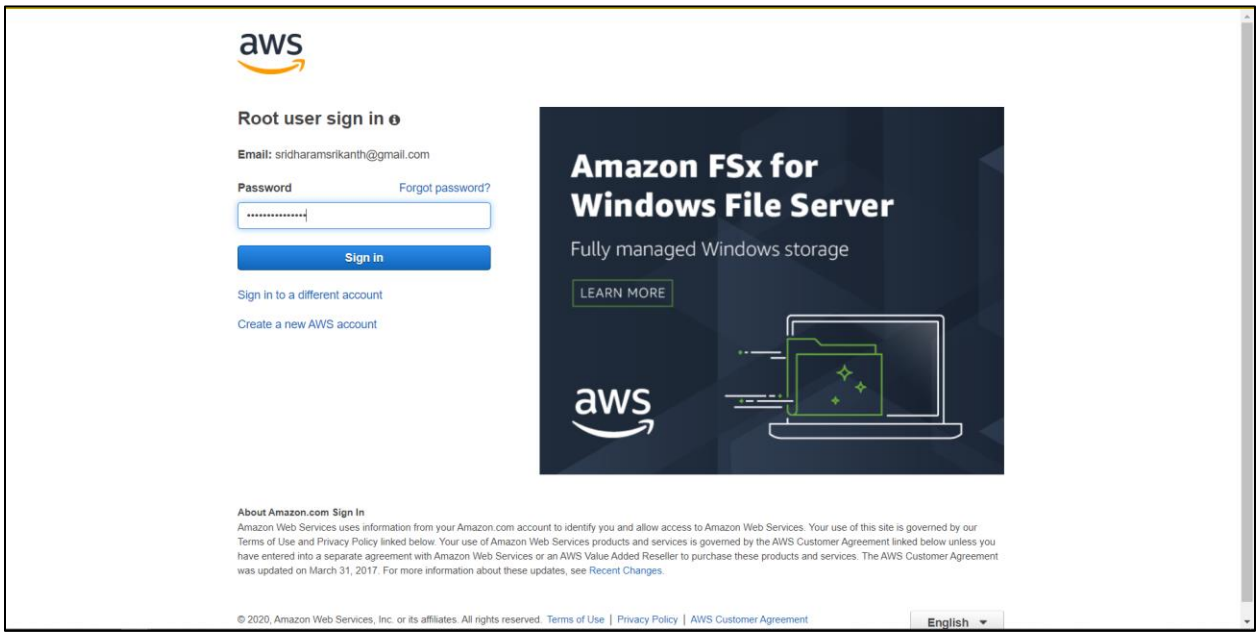
STEP 4 : - Creating S3 bucket.

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

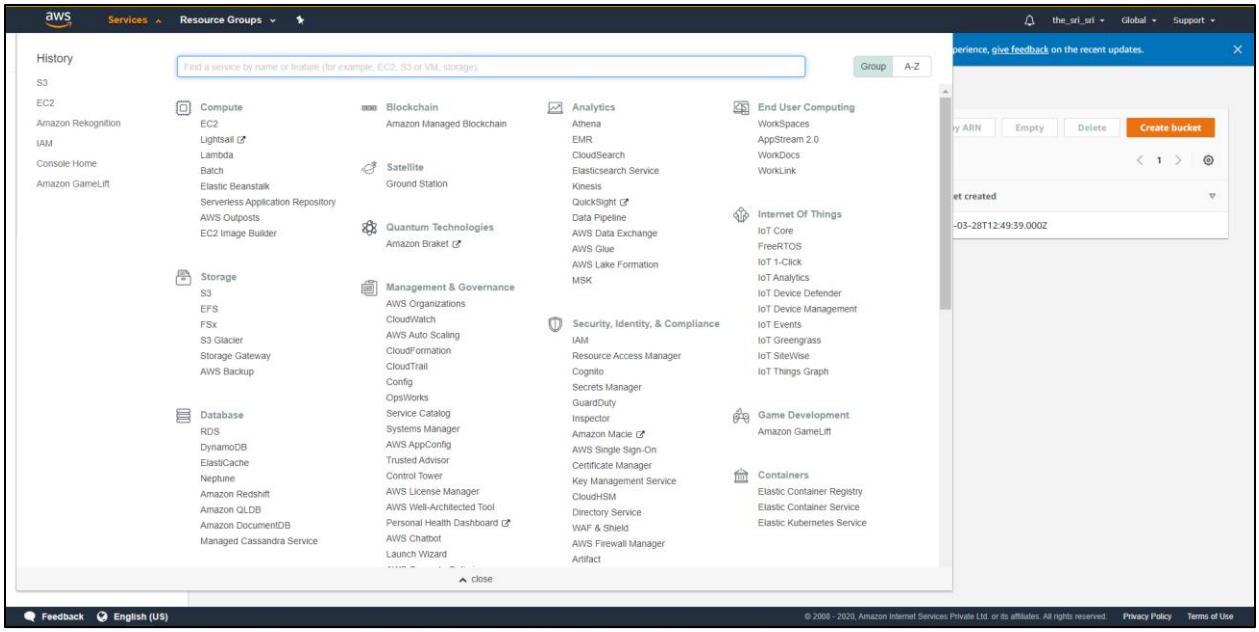
STEP 6 : - Tour to AWS rekognition services .

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.



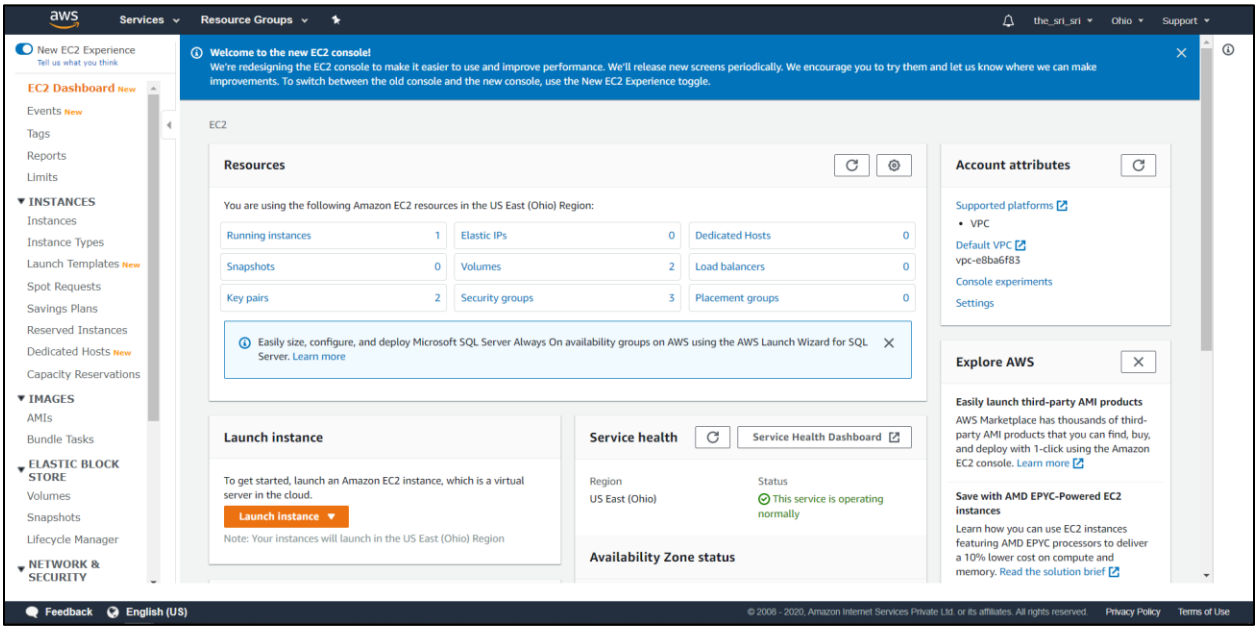
After logging in the dashboard of the services provided by Amazon AWS will be presented as below partitioned by their service .



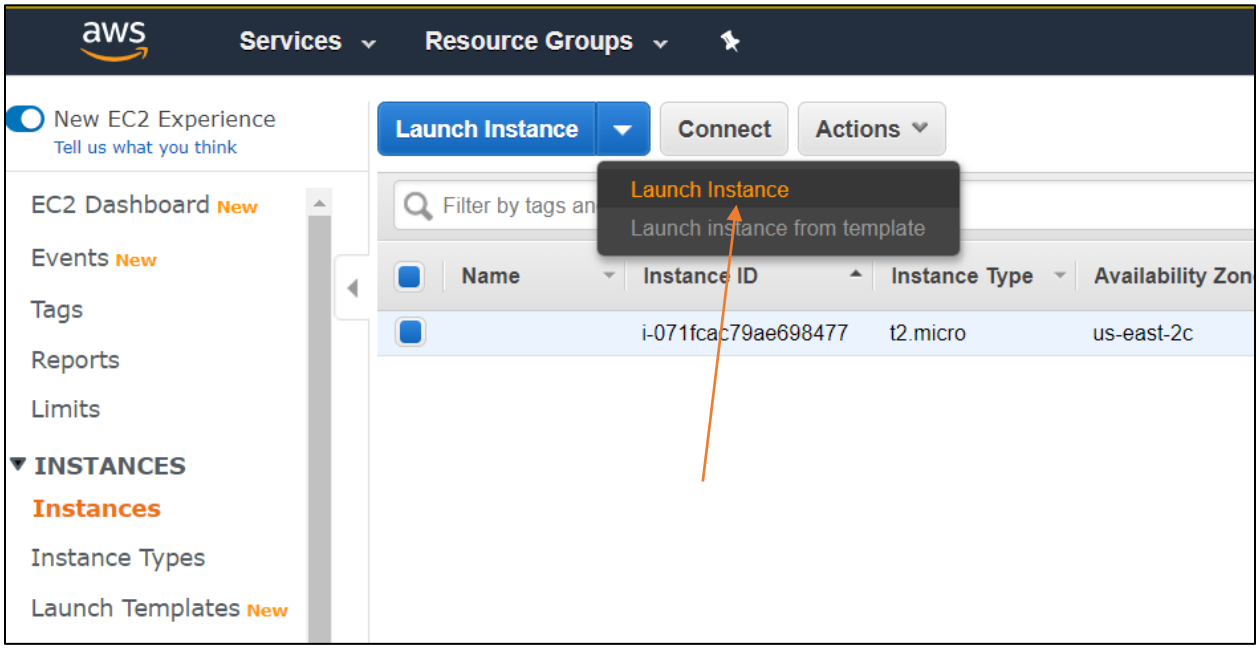
STEP 2 : - create an EC2 instance . follow the steps under given to create an EC2 instance .

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

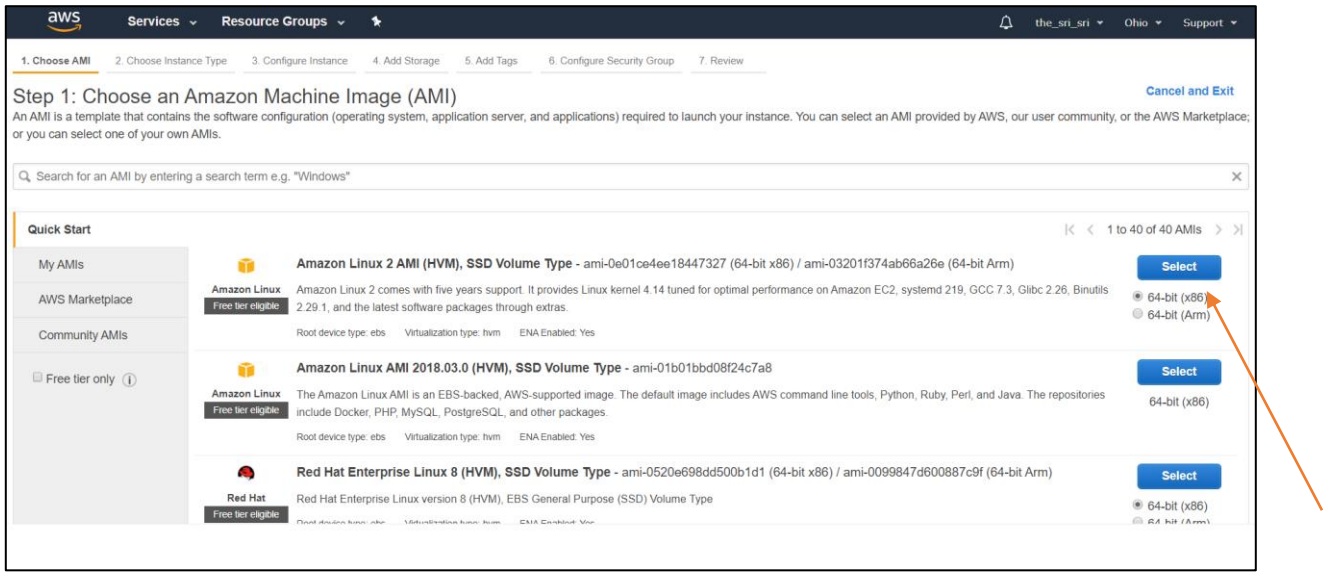
<u>EC2 dashboard</u>



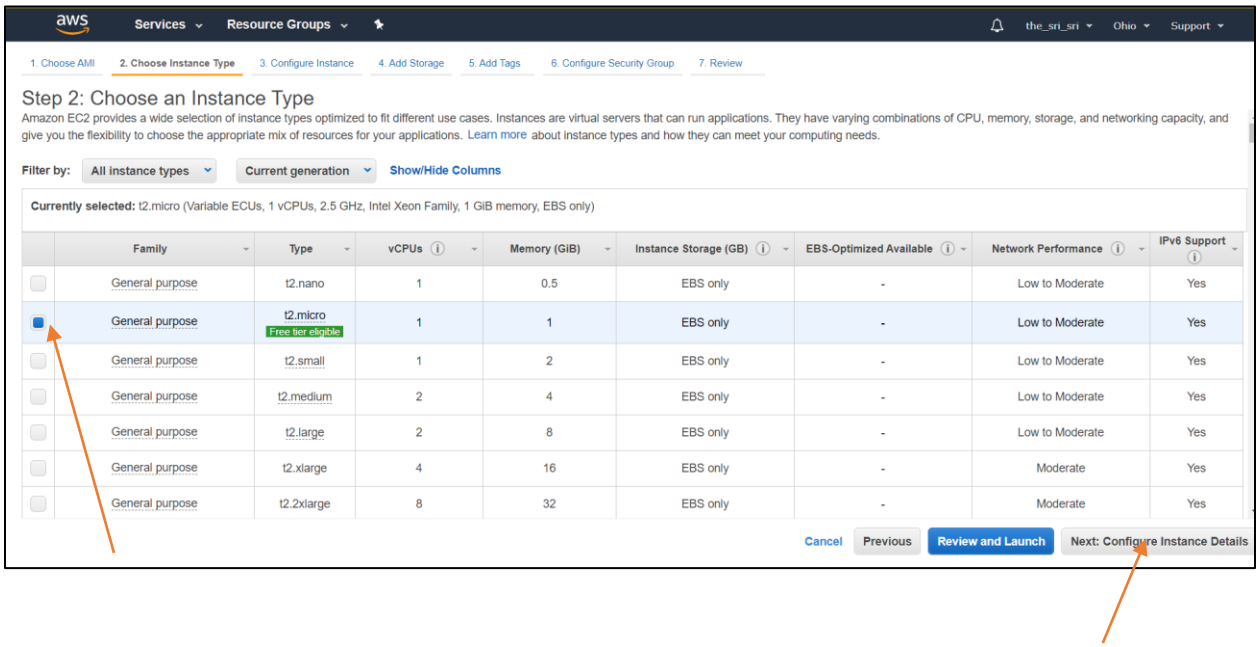
- select <u>Launch instance</u> under launch instance button .



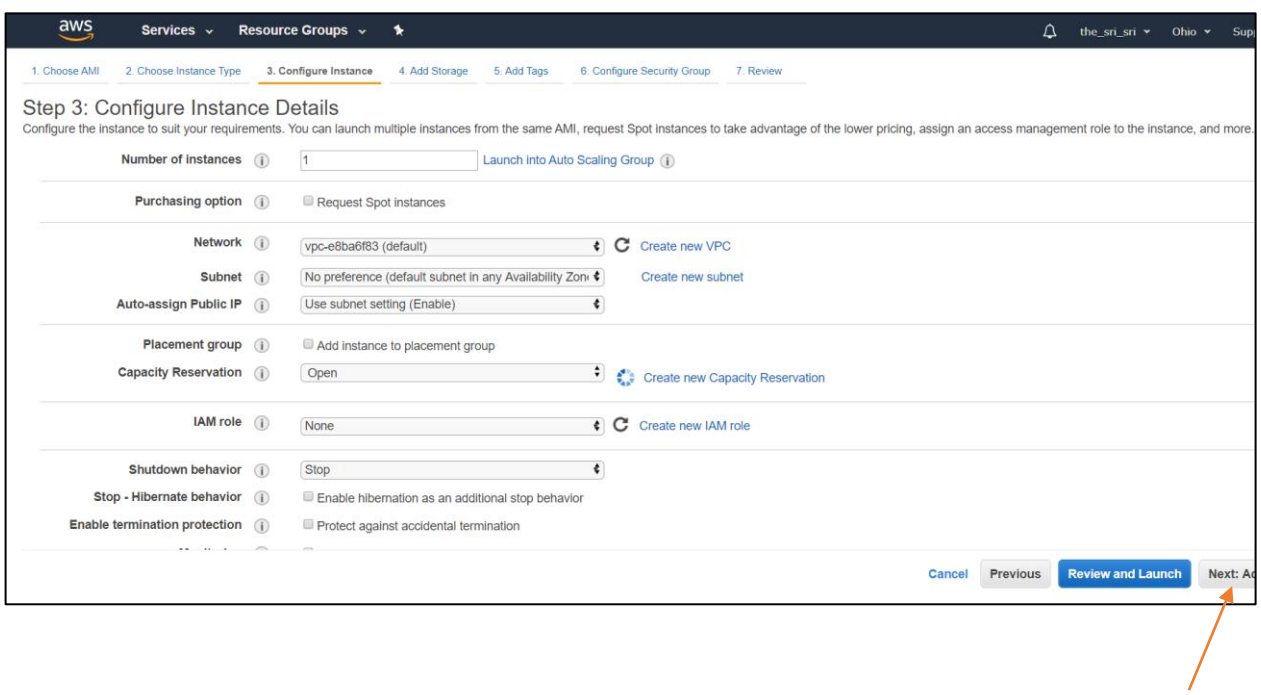
- Select an <u>Operating System</u> to work upon (here we have selected **Amazon Linux 2 Ami**)



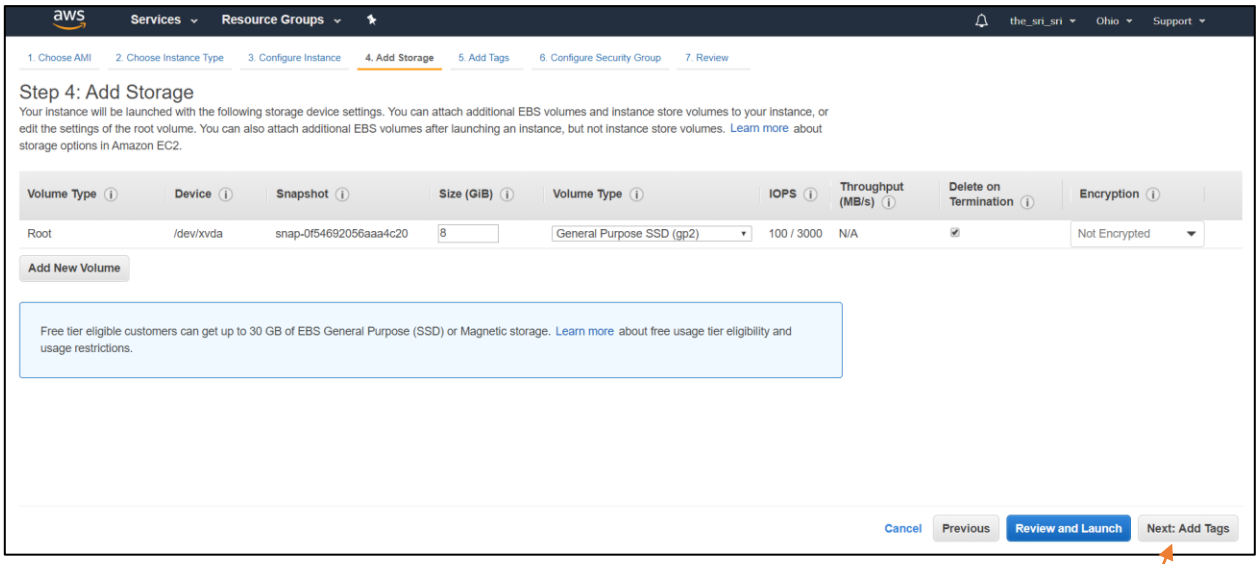
- Select required Instance type i.e. Type & No. of CPU , Memory needed based on your project needs. (for this project we have selected **t2.micro**) Now click **Next** .



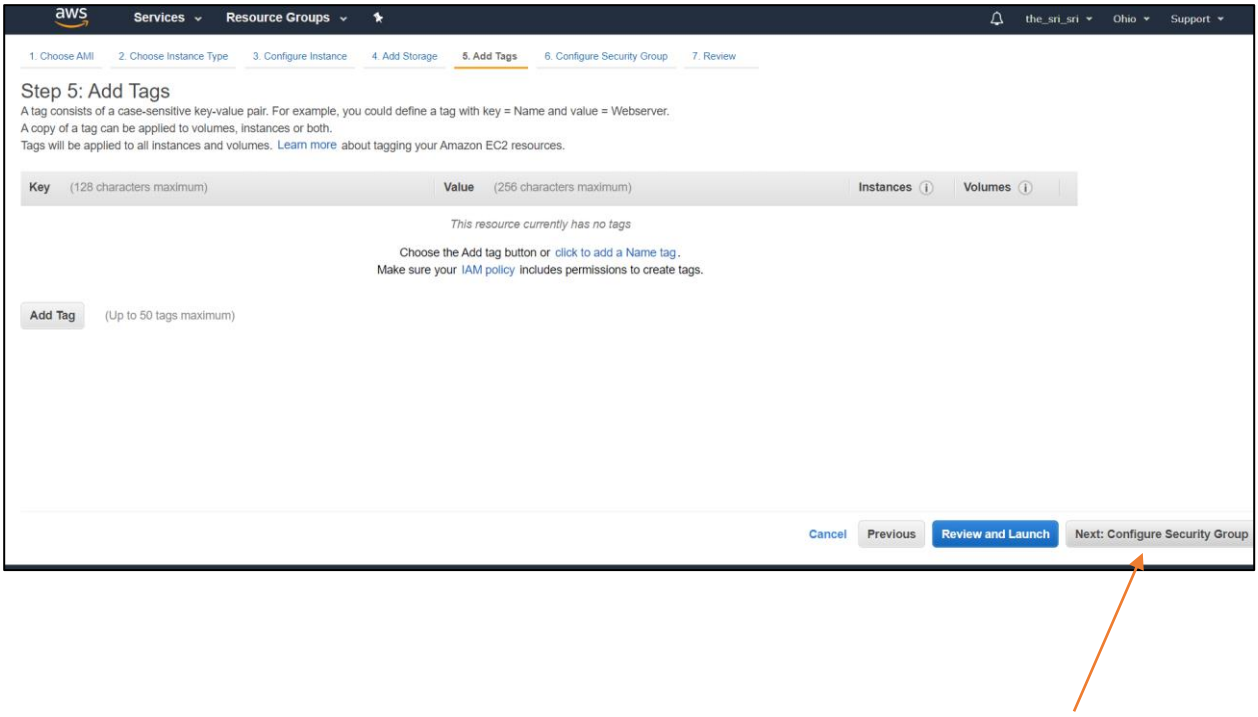
- Now Configure your Subnets in this step according to your needs (we **haven't done** any changes as per our project) click **next.**



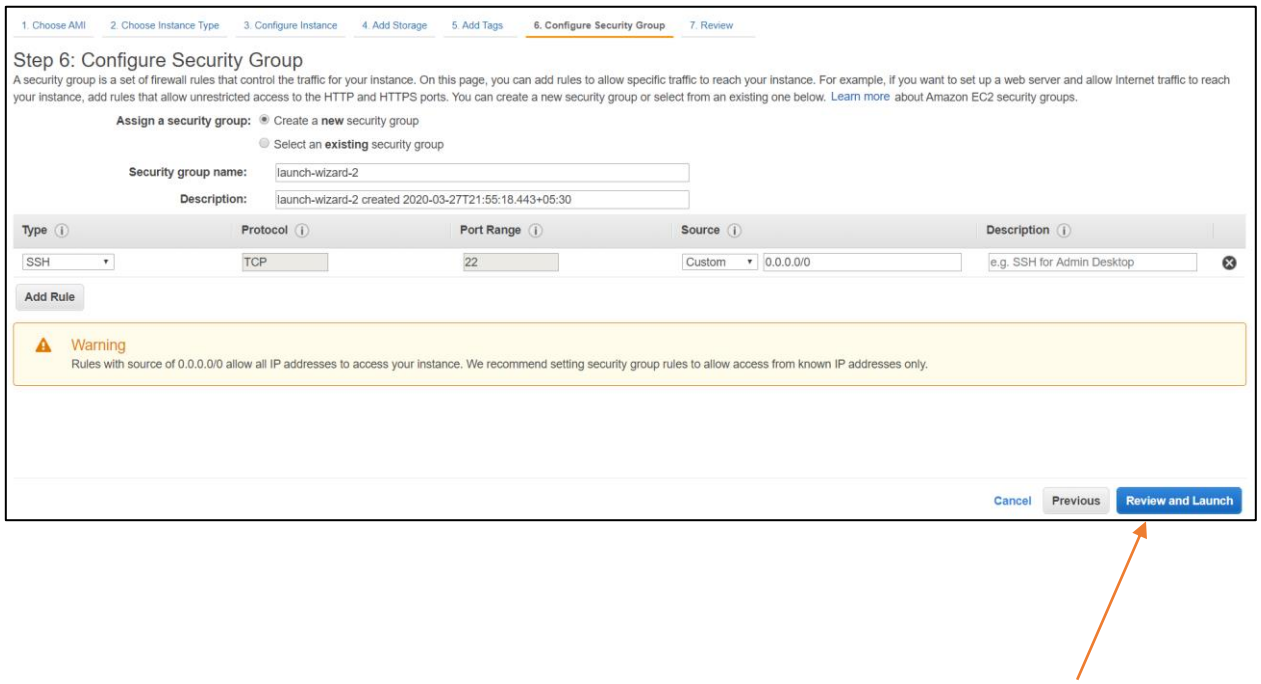
- Add Storage Details i.e. size of storage you need for your project (By default its 8GiB we **haven't done** any changes) click **Next.**



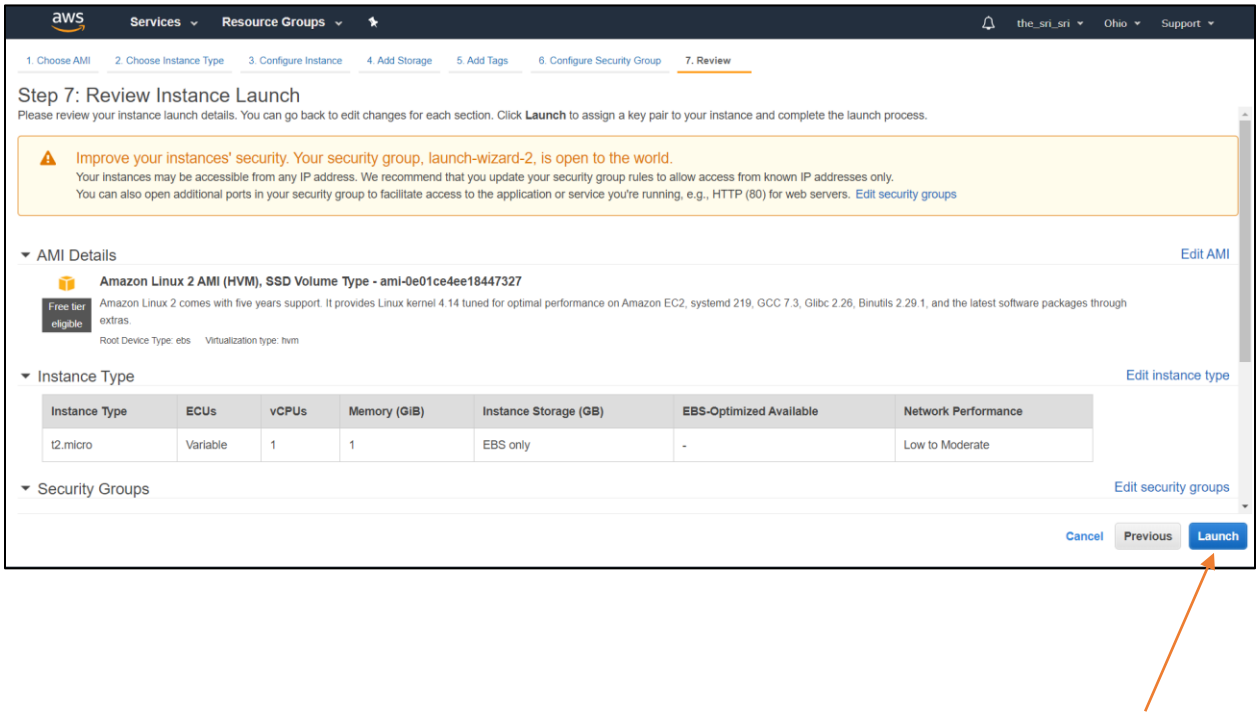
- here we **haven't added** any tags for our EC2 instance just click **NEXT.**



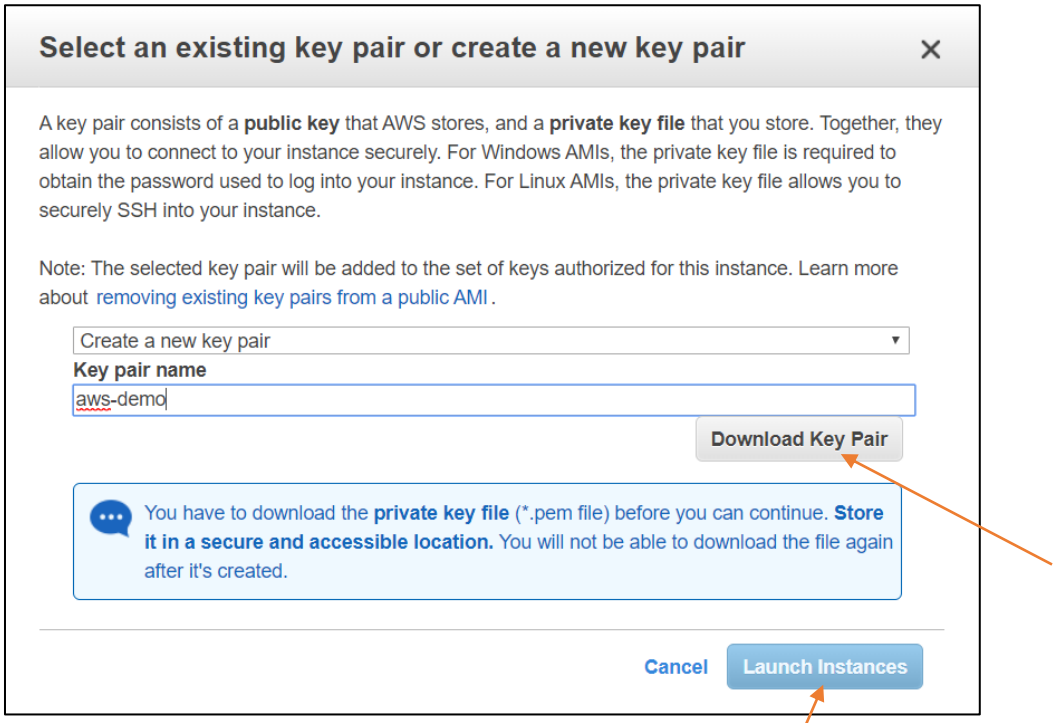
- Configure your security group according to the Type of EC2 we have created .(we have created **SSH type** i.e. **Linux based OS** and for that **port No.** is **22**) click **PREVIEW & LAUNCH.**



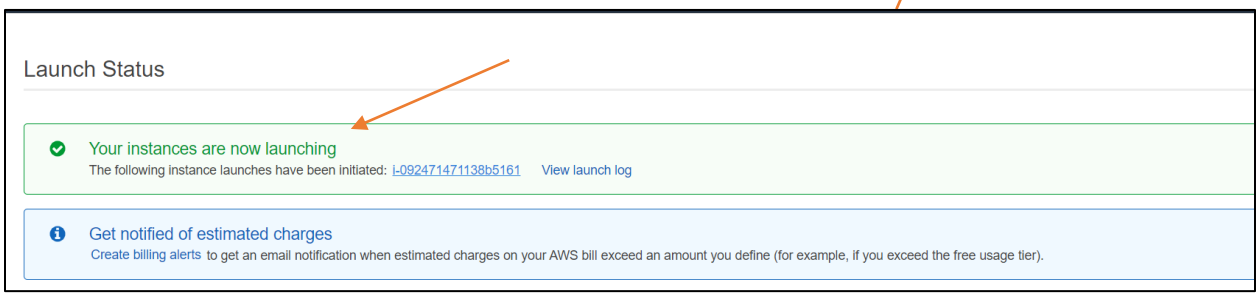
- In this step **preview** all your **changes and configurations** for confirmation and then click **launch.**



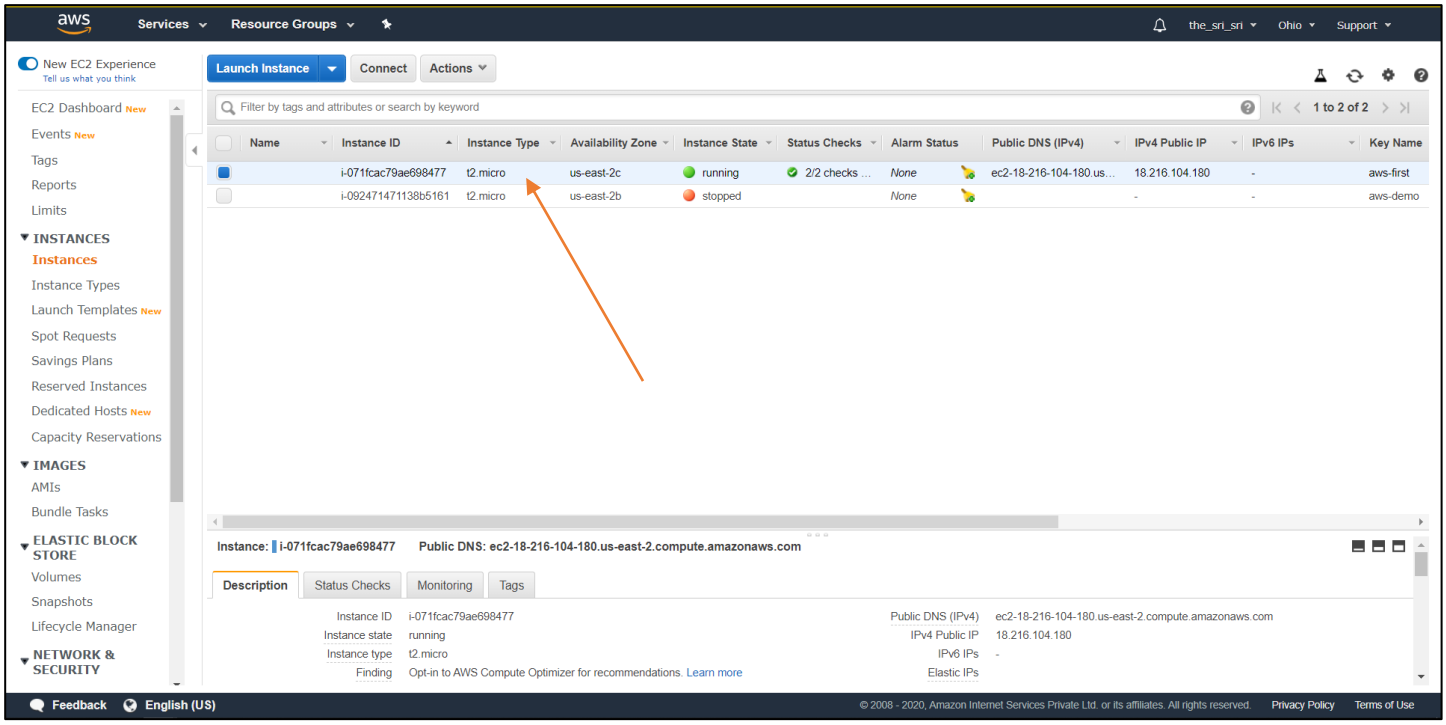
- On clicking **LAUNCH** a dialogue box will appear for generating private Key for the instance. Select **Create a new key Pair** option from the **drop down** and provide **a name** of your choice for your key . then **download key pair** and select **launch instance**.



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



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



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

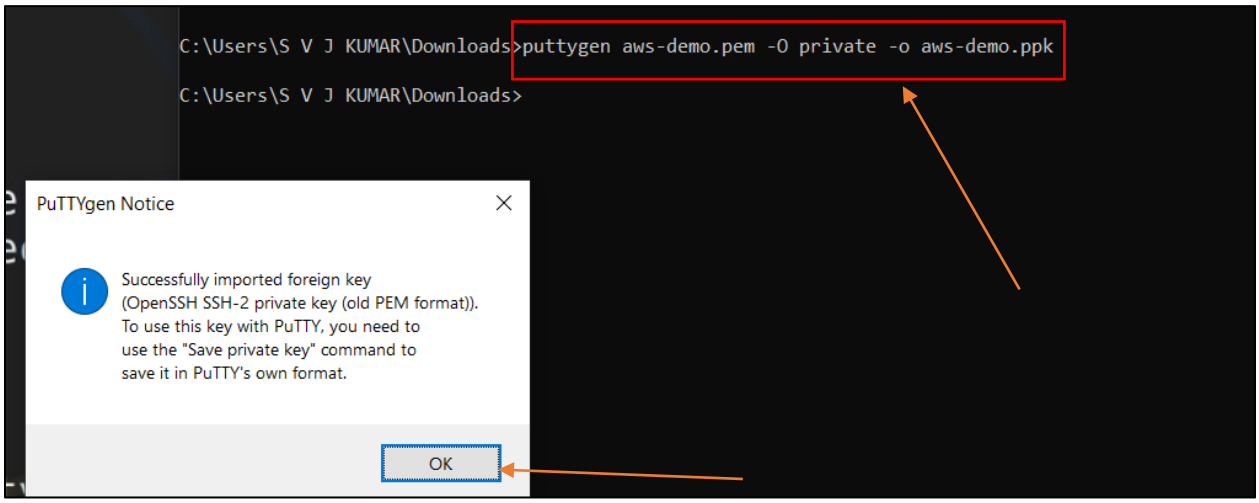
Now that we have created EC2 instance we should be connecting to this Instance using **PuTTY**.

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.

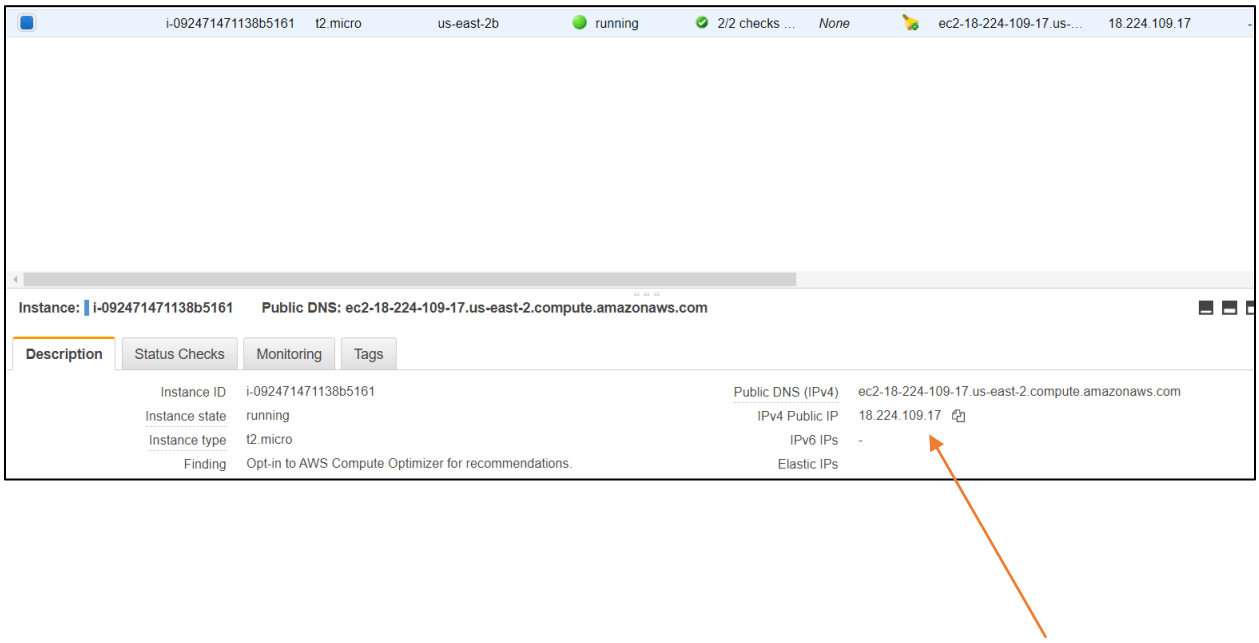
In the process of launching our instance we have downloaded a **Key file** which is in .pem format and for this key file to be used with PuTTY we need to convert is to .pmk format, we will be doing this using **keygen** command in our windows **CMD prompt** .



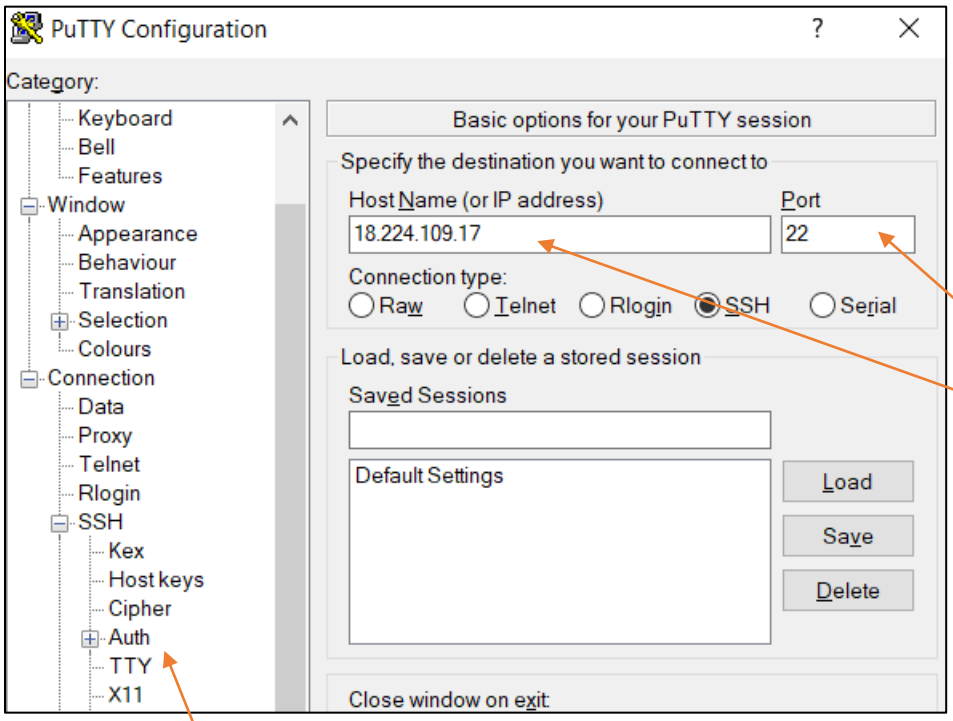
Type the **highlighted command** in the cmd prompt and click ok to get .ppk format of your **key file** .

After downloading and installing the PuTTY launch it .

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

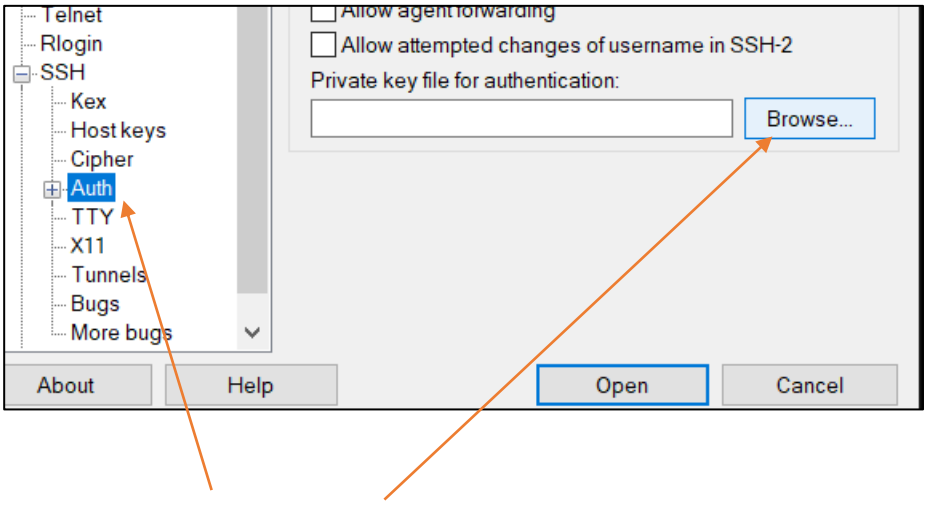


And **paste it** on the **PuTTY Configuration GUI.**

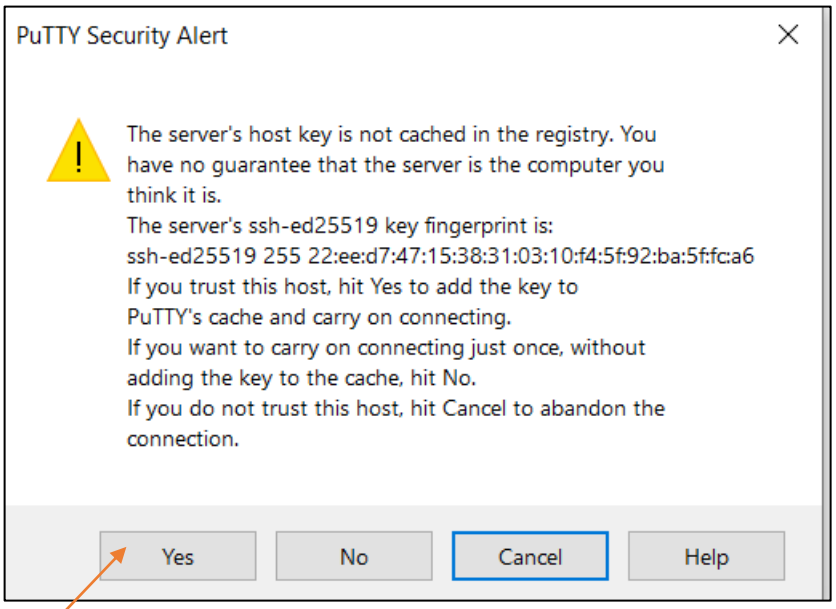


Paste your IP address and make sure the port No. is 22 and the connection type is SSH.

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.

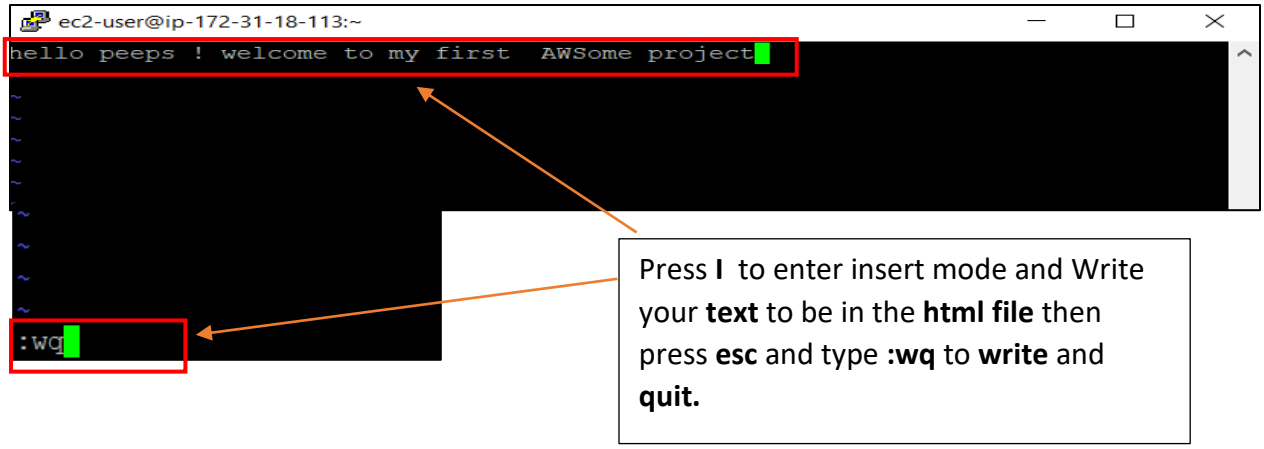


Select **yes** for this security alert dialogue box. And your will be seeing an **Command Line Interface** of **PuTTY .**

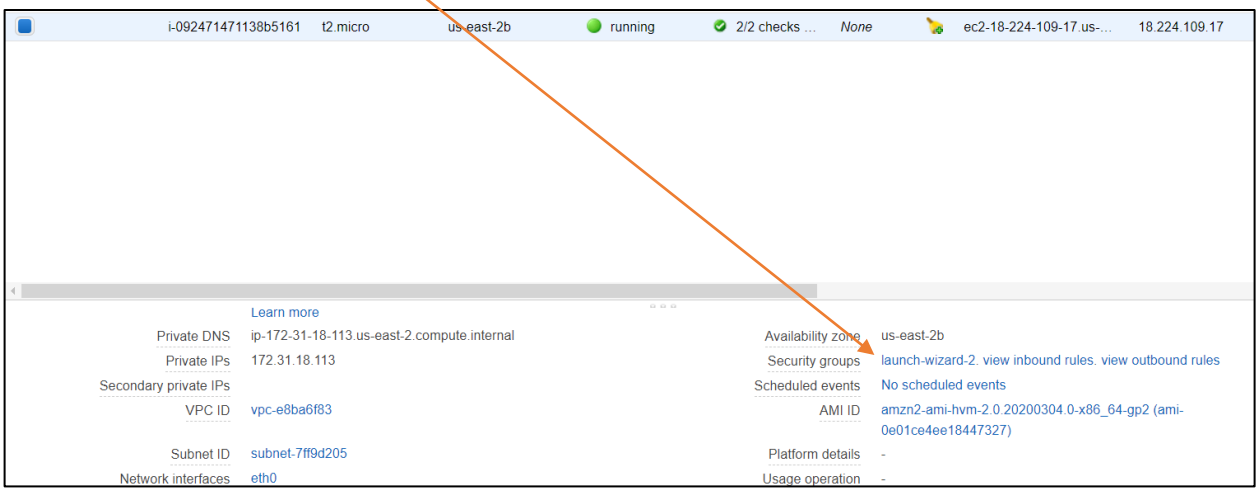
Type **the sequence of commands** in the command line interface of **PuTTY to connect** to **EC2 instance** .

```
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
```

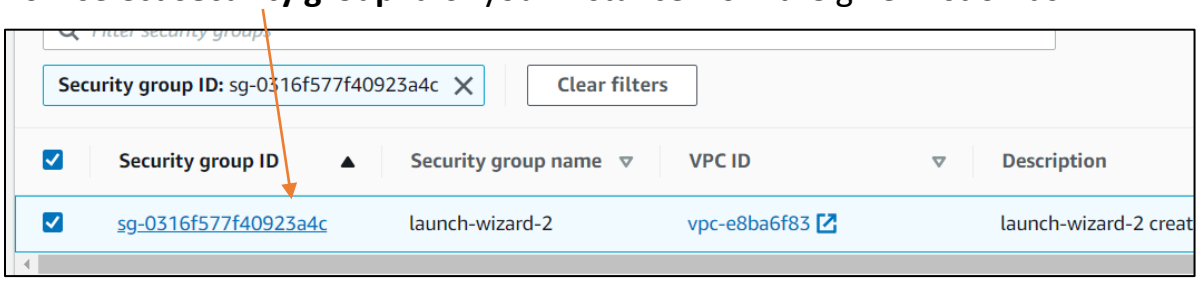
```
ec2-user@ip-172-31-18-113:~
[ec2-user@ip-172-31-18-113 ~]$ sudo service httpd start
Redirecting to /bin/systemctl start httpd.service
[ec2-user@ip-172-31-18-113 ~]$ sudo service httpd status
Redirecting to /bin/systemctl status httpd.service
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/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
```



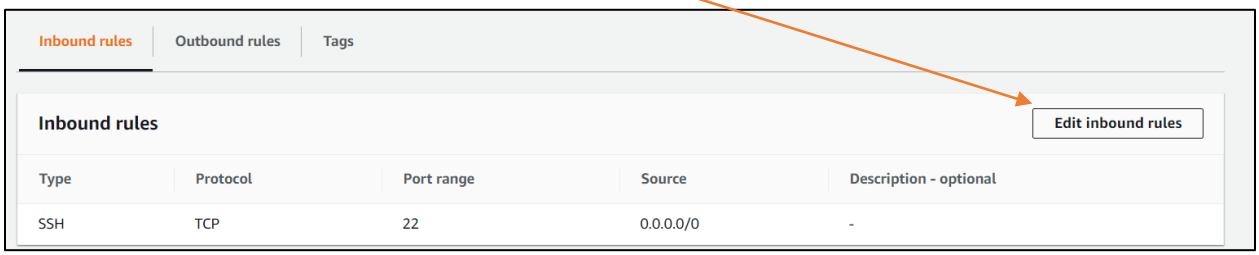
Now go back to the **EC2 instance dashboard** and **select your instance** before selecting **launch-wizards-2** from the **security groups** tag.



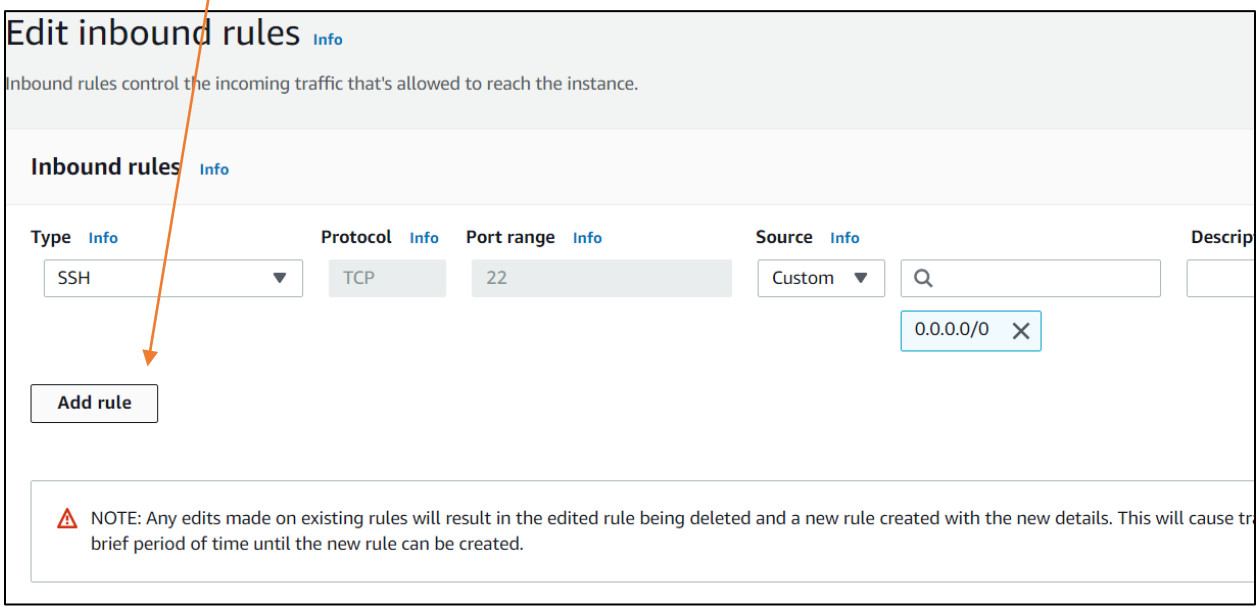
now select **security group id** of your instance from the given list of ids

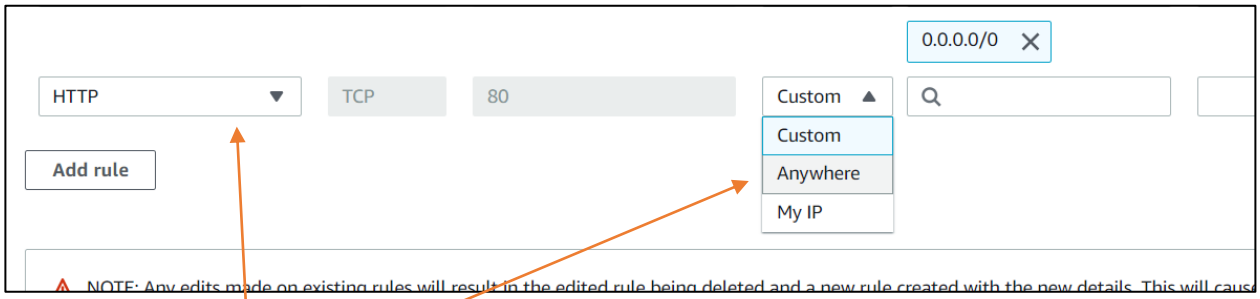


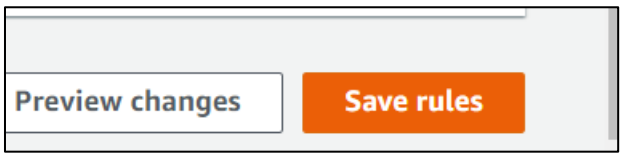
now select **Edit inbound rules .**



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** .

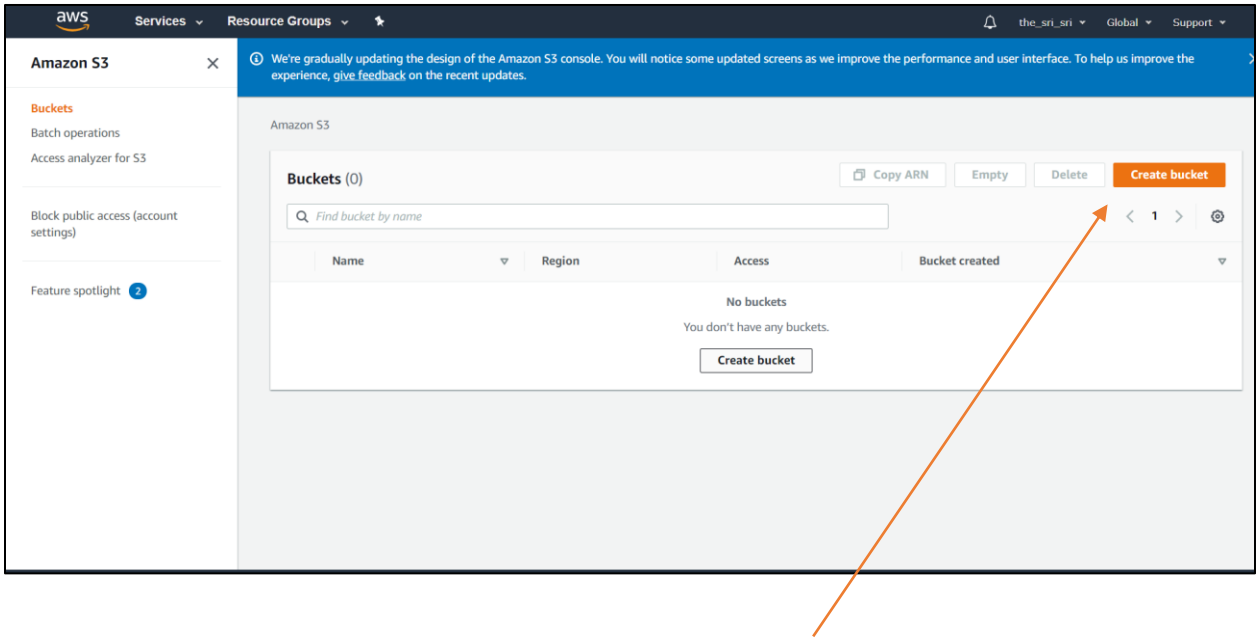
now search for your **IP address** in your **web browser** and watch your static **web page** on the internet **live.**

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.

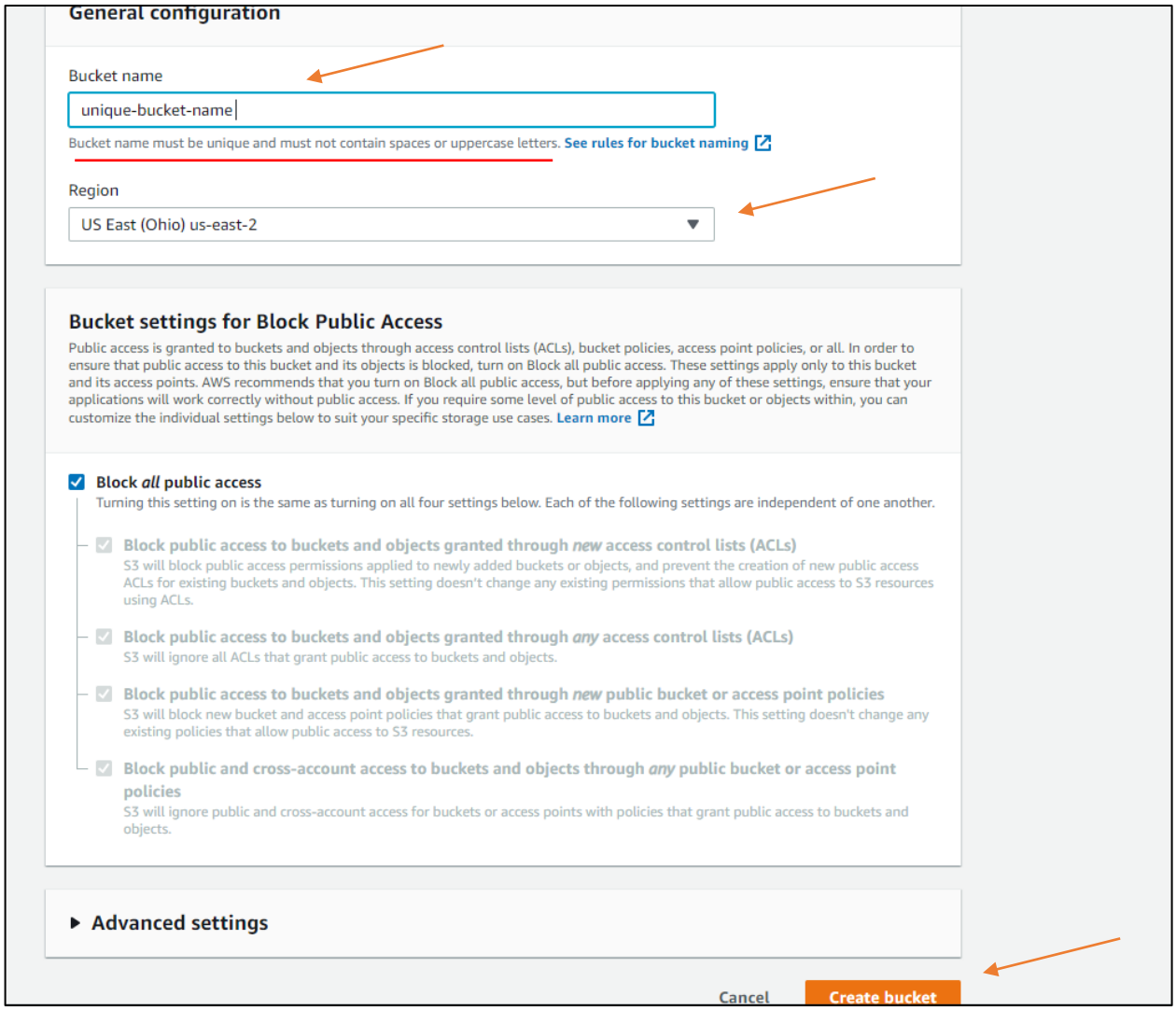
Buckets do not have any region it agrees to be global unlike the objects in the bucket which are accessible from specified region only .

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

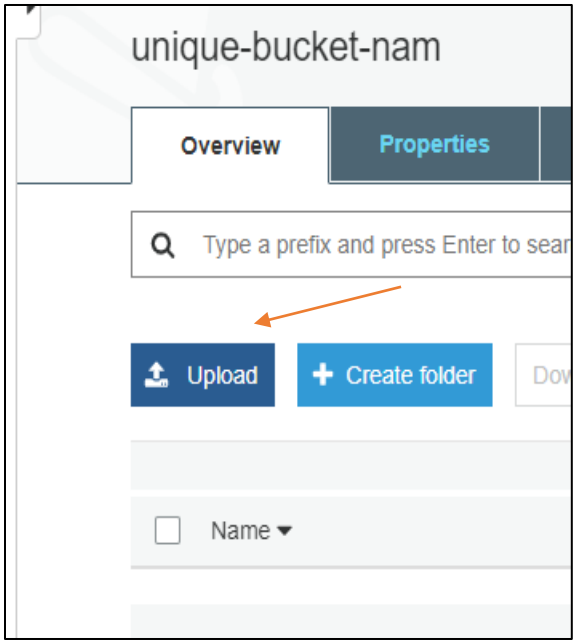


To create your first bucket start your steps by clicking the **"Create Bucket"** button on the right corner of the **Dashboard.**

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'**.



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



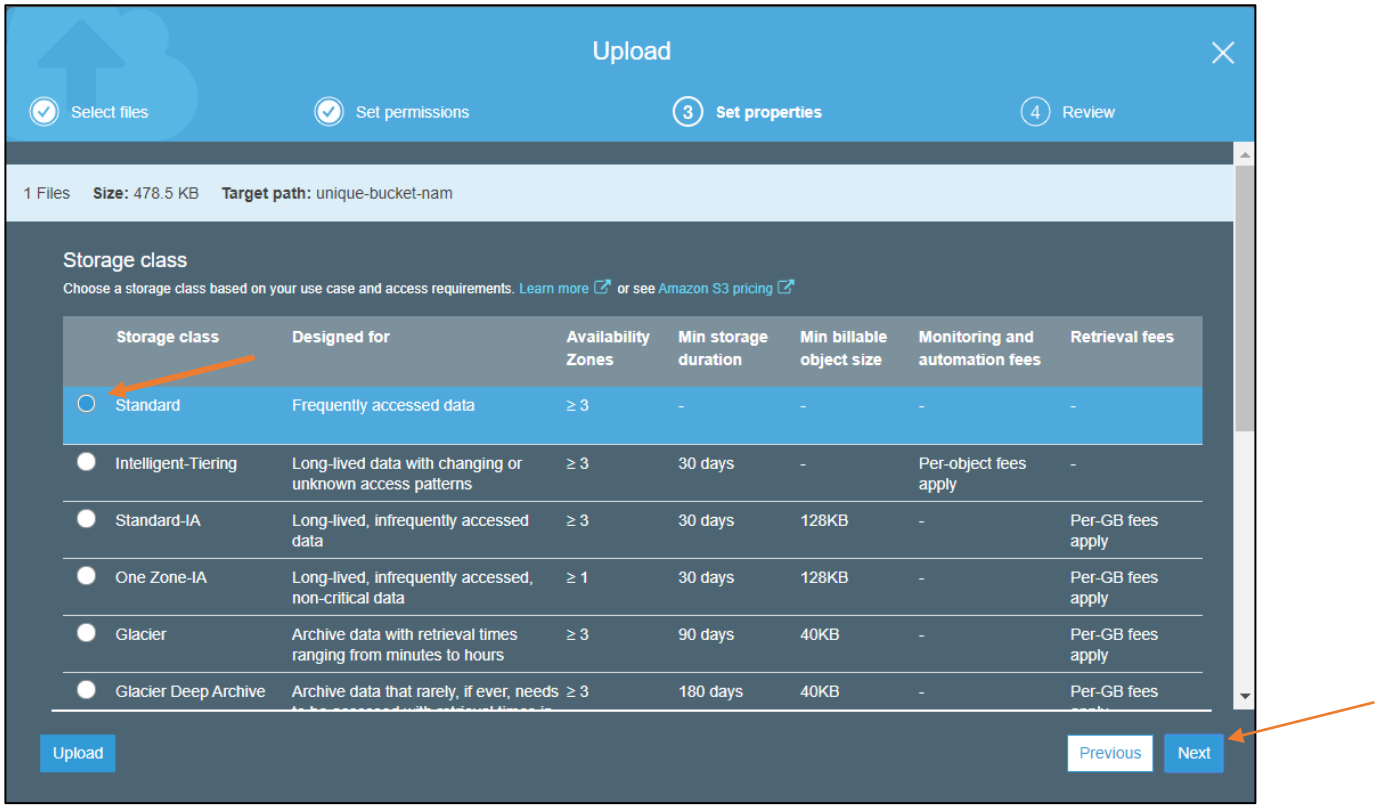
Or you can create **folder** to make **collections** of your **objects.**

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.**

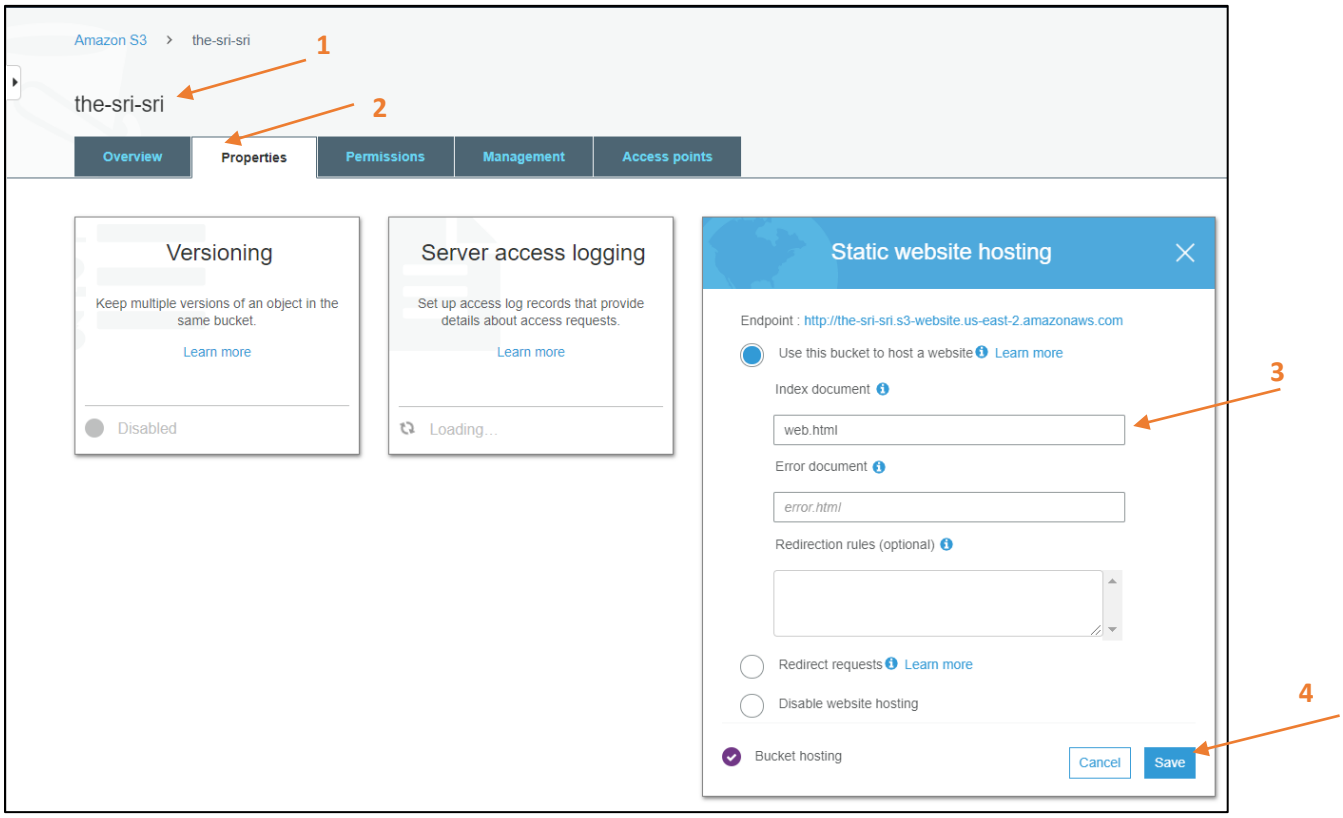
1. Standard : Data **accessed frequently** are stored in all the **Availability zones** of the **region** you selected earlier.

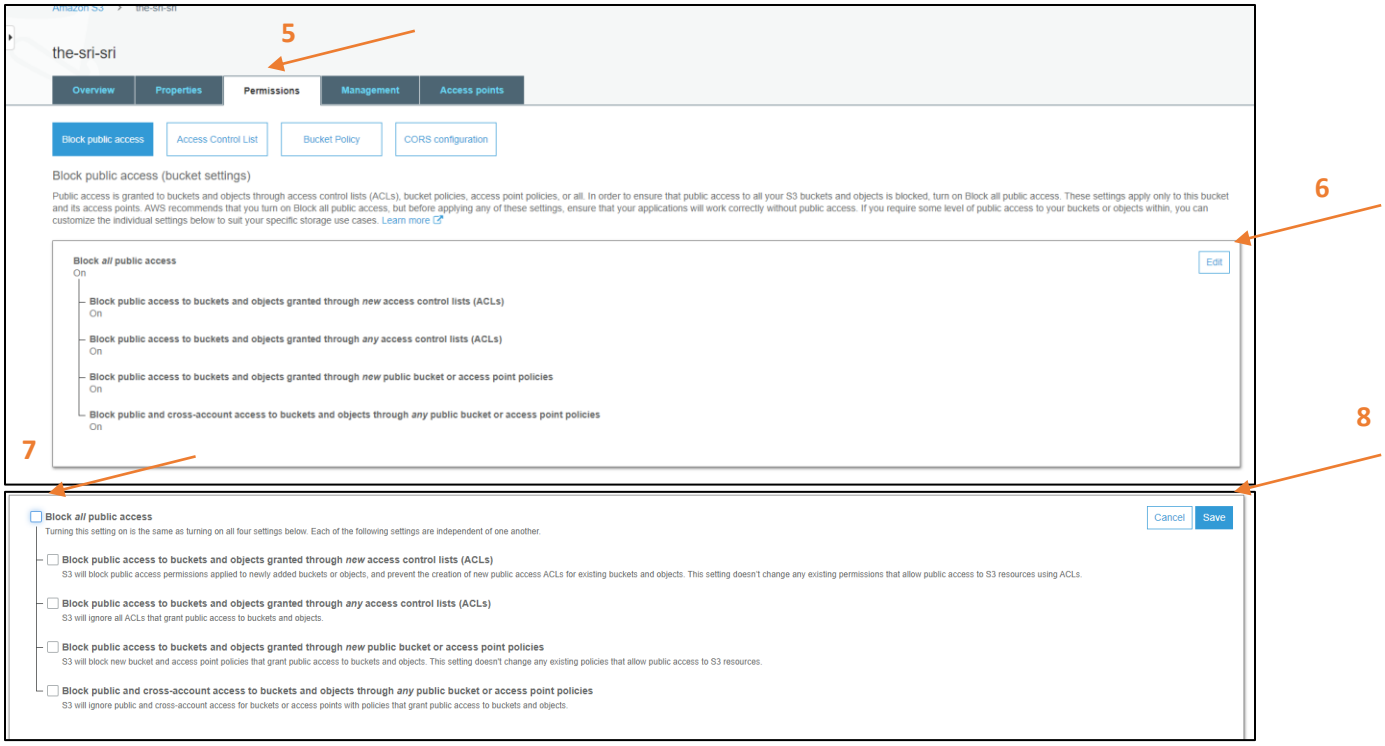


2. Intelligent-tiering : for the data to be accessed with unknown patterns . but critical / Important.
3. Standard-IA : data to be stored for a long time , infrequently accessed , and non-critical.
4. Glacier : archived data , very rarely accessed , takes more time for retrieval.
5. Glaciers Deep Archive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

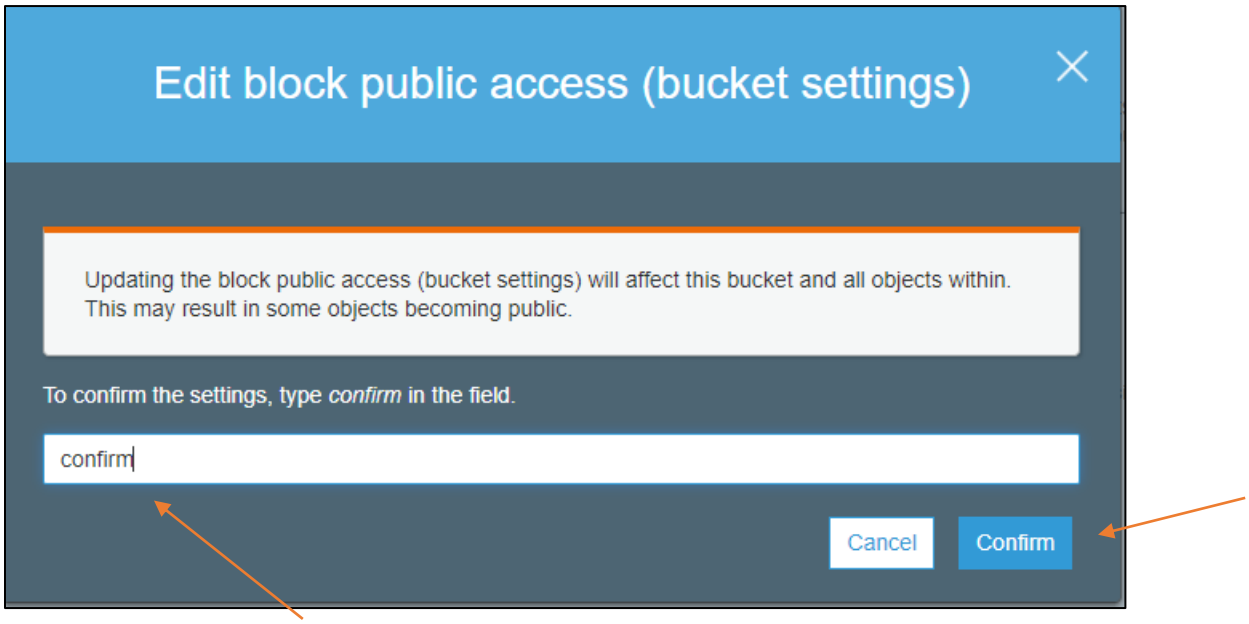
Follow under given steps to Host your file to the web.

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**

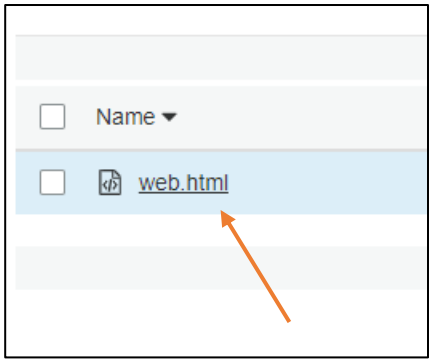


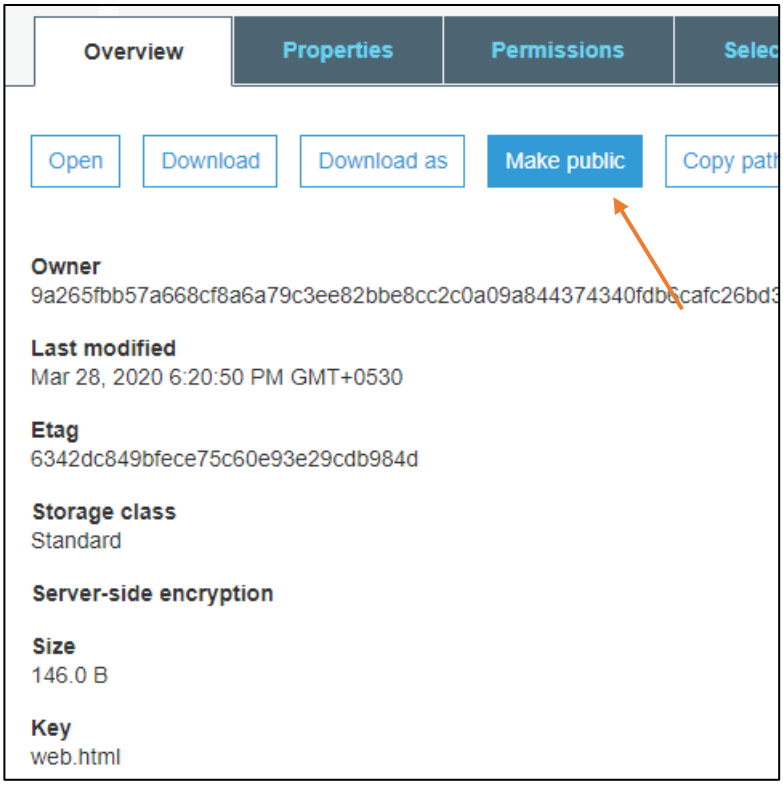


9. Type **"Confirm"** in the dialogue box appeared and press **confirm.**



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


Name
web.html

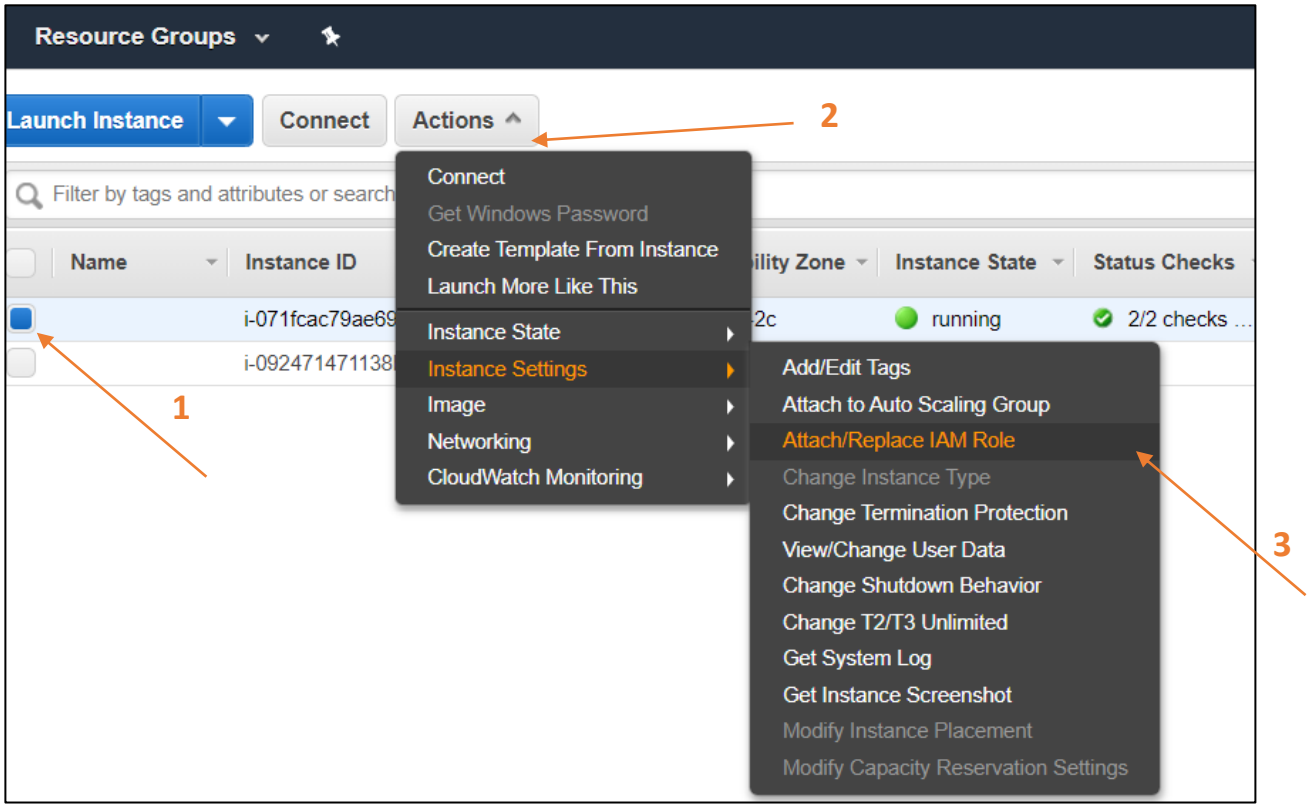

Overview
Properties
Permissions
Open
Download
Download as
Make public
Owner
9a265fbb57a668cf8a6a79c3ee82bbe8cc2c0a09a844374340fdb6cafc26bd3
Last modified
Mar 28, 2020 6:20:50 PM GMT+0530
Etag
6342dc849bfece75c60e93e29cdb984d
Storage class
Standard
Server-side encryption
Size
146.0 B
Key
web.html

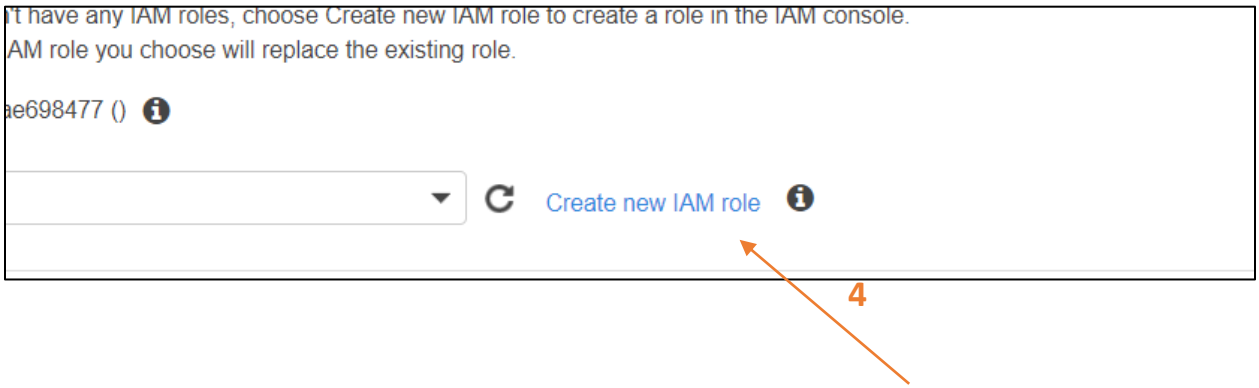

Not secure | the-sri-sri.s3-website.us-east-2.amazona
Apps
7 Layers of The OSI...
(17) YouTube
C program to swap...
SRIDHARAM SRIKANTH
This is my first web page hosted live on web

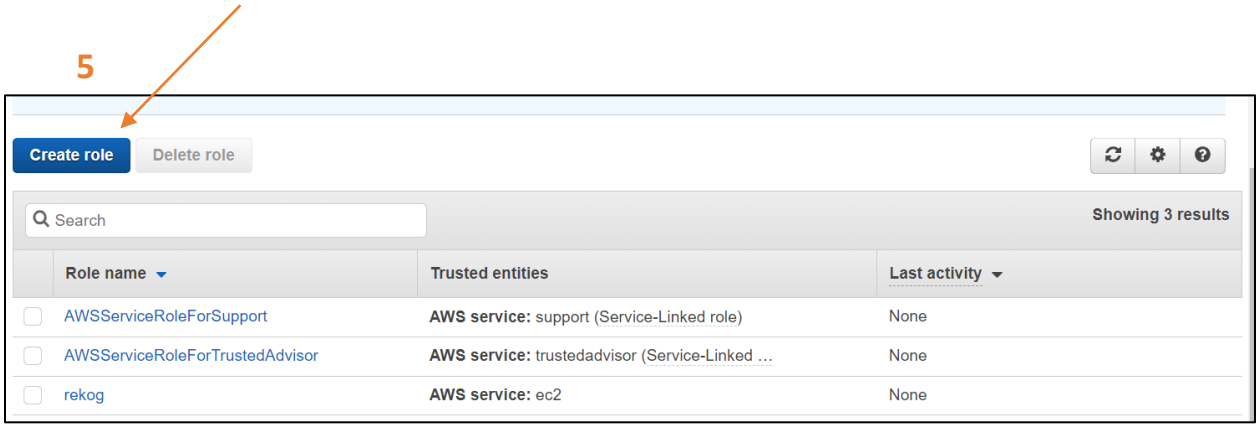
STEP 5 : - Connecting S3 to EC2 and uploading Objects to S3

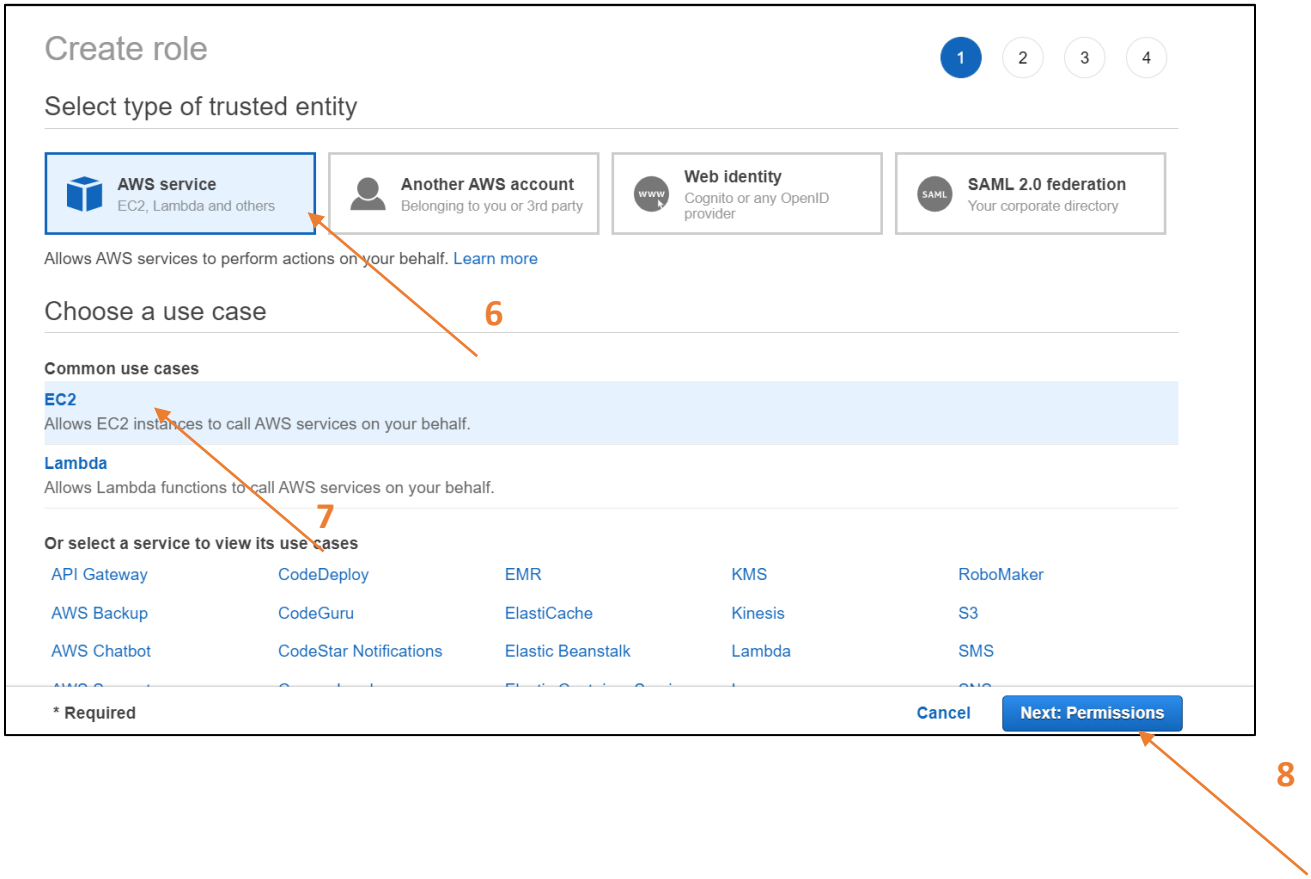
Create **IAM Role** .

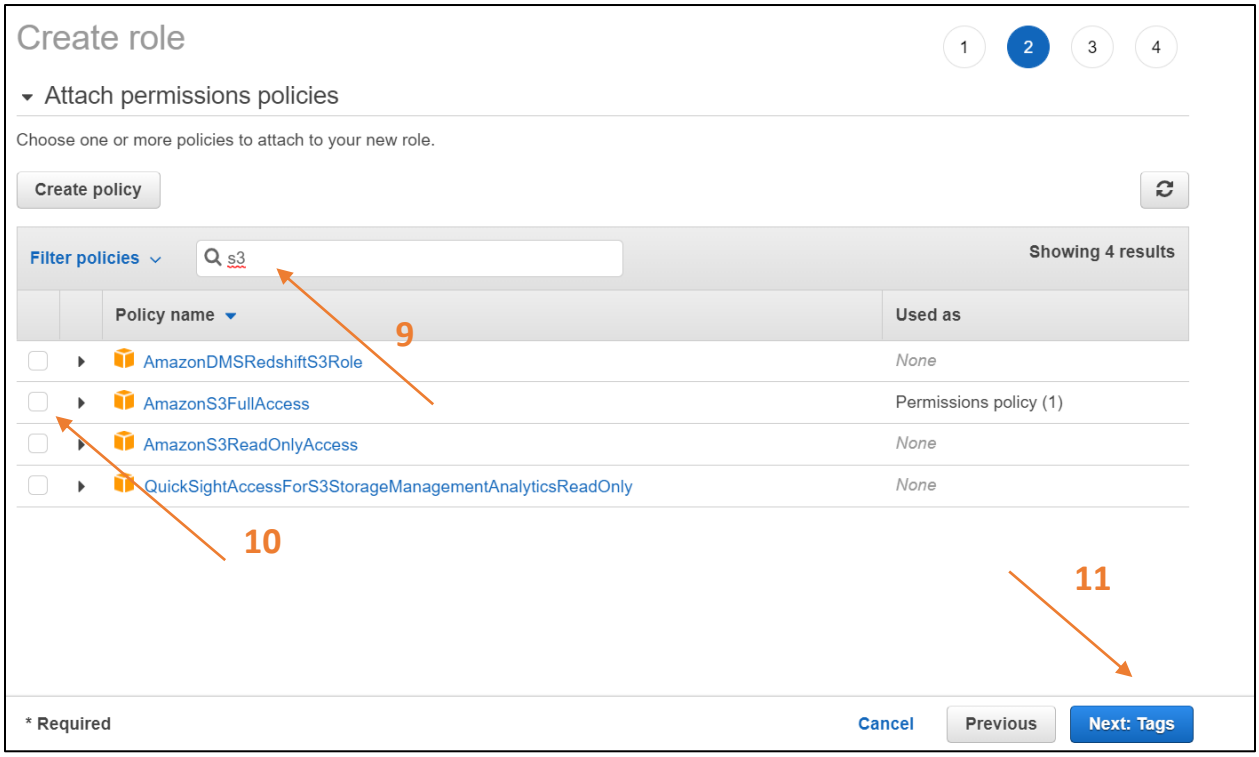
Select your EC2 instance and then **follow bellow illustrated steps.**

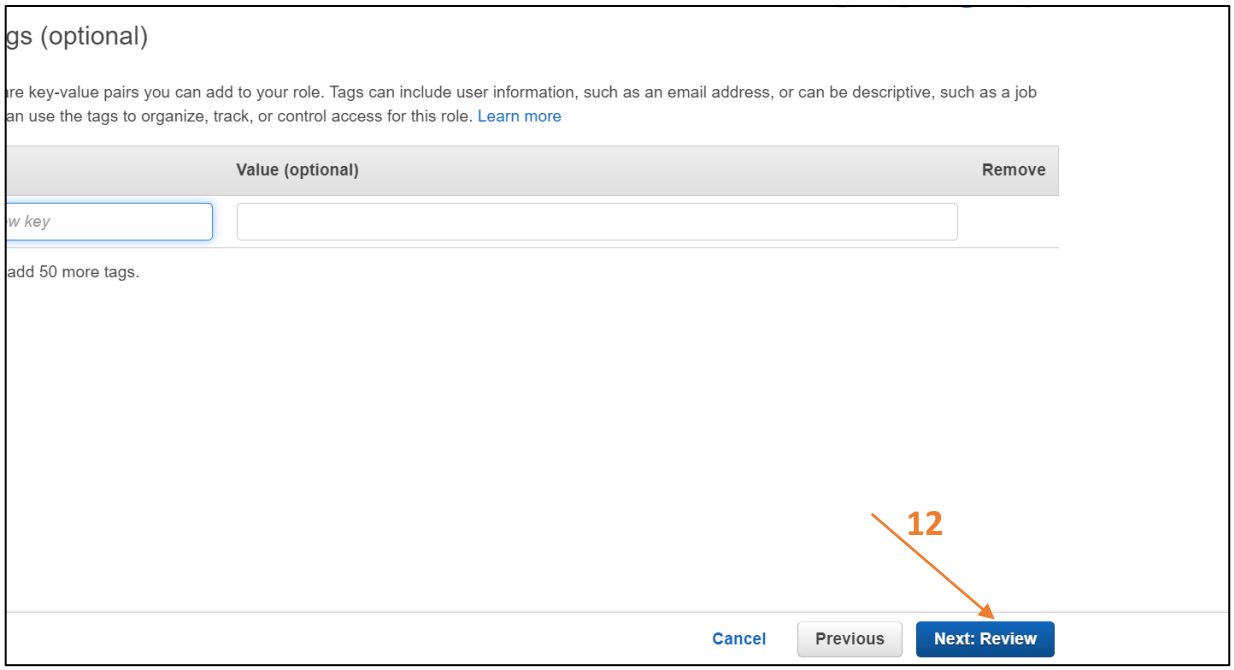


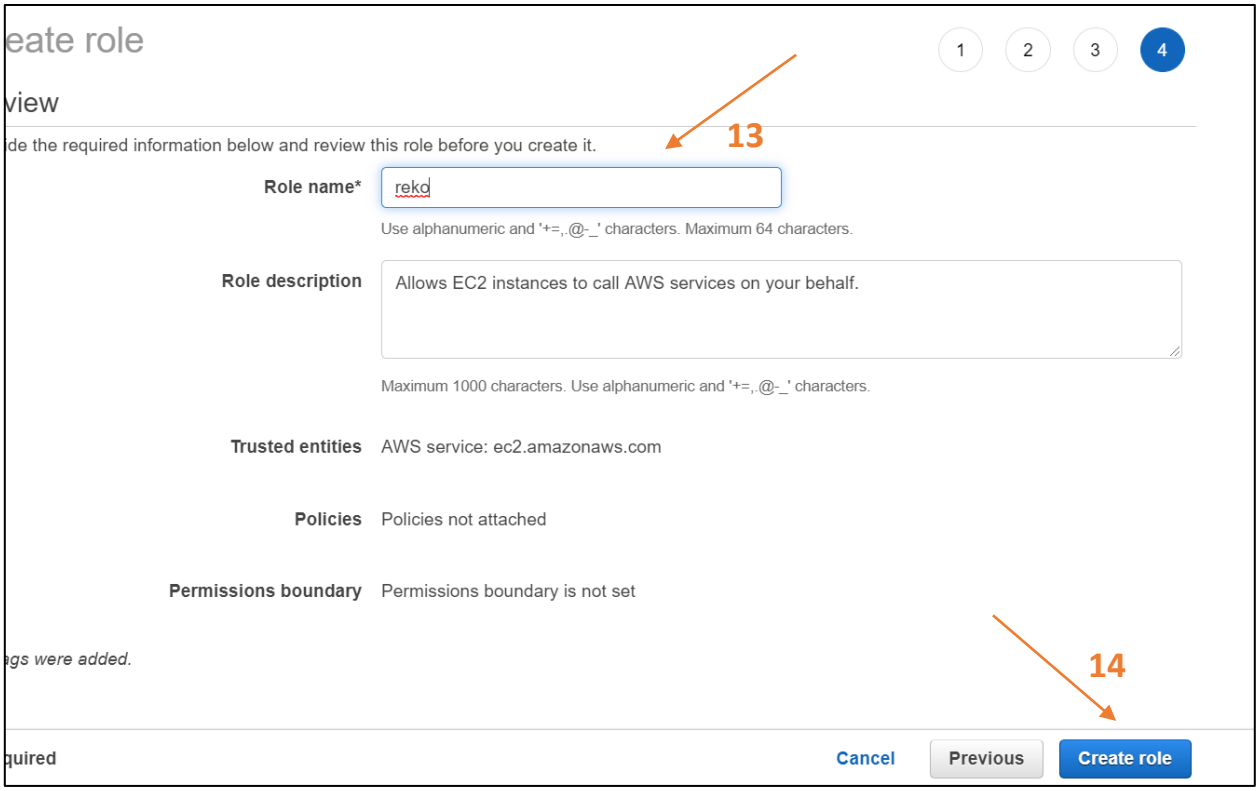



5
Create role
Delete role
Search
Showing 3 results
Role name
Trusted entities
Last activity
AWSServiceRoleForSupport
AWS service: support (Service-Linked role)
None
AWSServiceRoleForTrustedAdvisor
AWS service: trustedadvisor (Service-Linked ...
None
rekog
AWS service: ec2
None

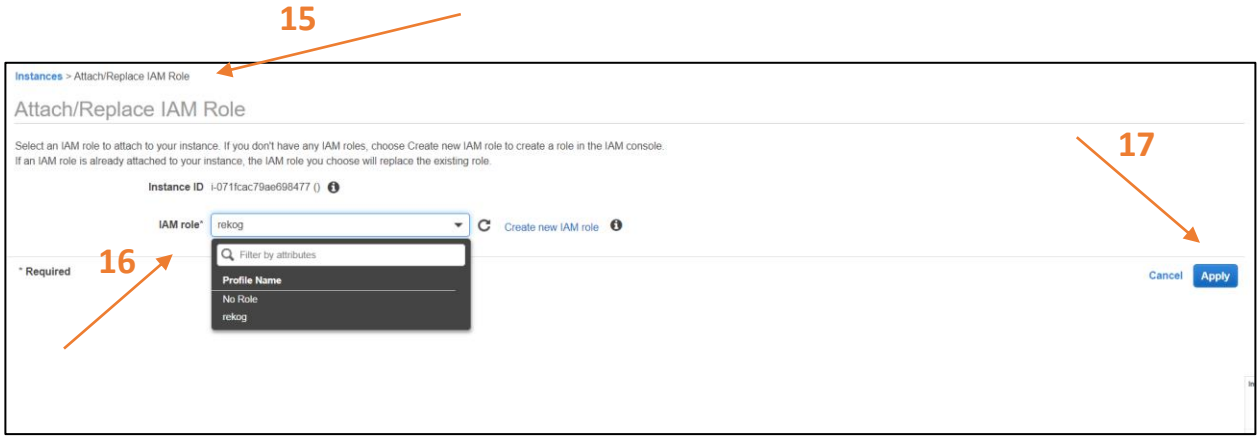

Create role
1
2
3
4
Select type of trusted entity
AWS service
EC2, Lambda and others
Another AWS account
Belonging to you or 3rd party
Web identity
Cognito or any OpenID provider
SAML 2.0 federation
Your corporate directory
Allows AWS services to perform actions on your behalf. Learn more
Choose a use case
6
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.
7
Or select a service to view its use cases
API Gateway
CodeDeploy
EMR
KMS
RoboMaker
AWS Backup
CodeGuru
ElastiCache
Kinesis
S3
AWS Chatbot
CodeStar Notifications
Elastic Beanstalk
Lambda
SMS
* Required
Cancel
Next: Permissions
8


Create role
1
2
3
4
Attach permissions policies
Choose one or more policies to attach to your new role.
Create policy
Filter policies
s3
Showing 4 results
Policy name
Used as
AmazonDMSRedshiftS3Role
None
AmazonS3FullAccess
Permissions policy (1)
AmazonS3ReadOnlyAccess
None
QuickSightAccessForS3StorageManagementAnalyticsReadOnly
None
9
10
11
* Required
Cancel
Previous
Next: Tags


gs (optional)
re 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
an use the tags to organize, track, or control access for this role. Learn more
Value (optional)
Remove
w key
add 50 more tags.
12
Cancel
Previous
Next: Review



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



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.pha
r
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 re
quire 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] [--n
o-suggest] [--no-update] [--no-scripts] [--update-no-dev] [--update-with-depende
ncies] [--update-with-all-dependencies] [--ignore-platform-reqs] [--prefer-stabl
e] [--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-bf13-6e1aa794d8c2
[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]$
```

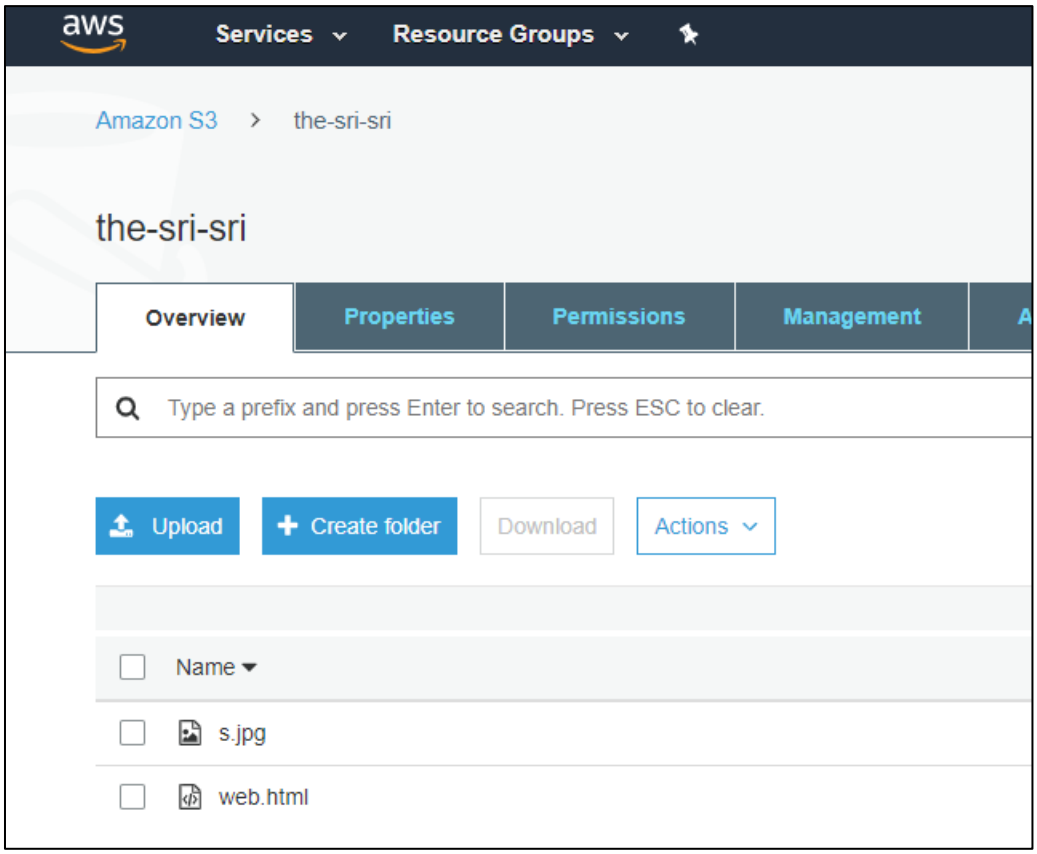
Sudo wget {link of image to be downloaded}

```
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/b97ea33b5842c78
94b804923c6c05580.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.jp
g
[ec2-user@ip-172-31-44-120 ~]$ ls
composer.phar  s.jpg
[ec2-user@ip-172-31-44-120 ~]$
```

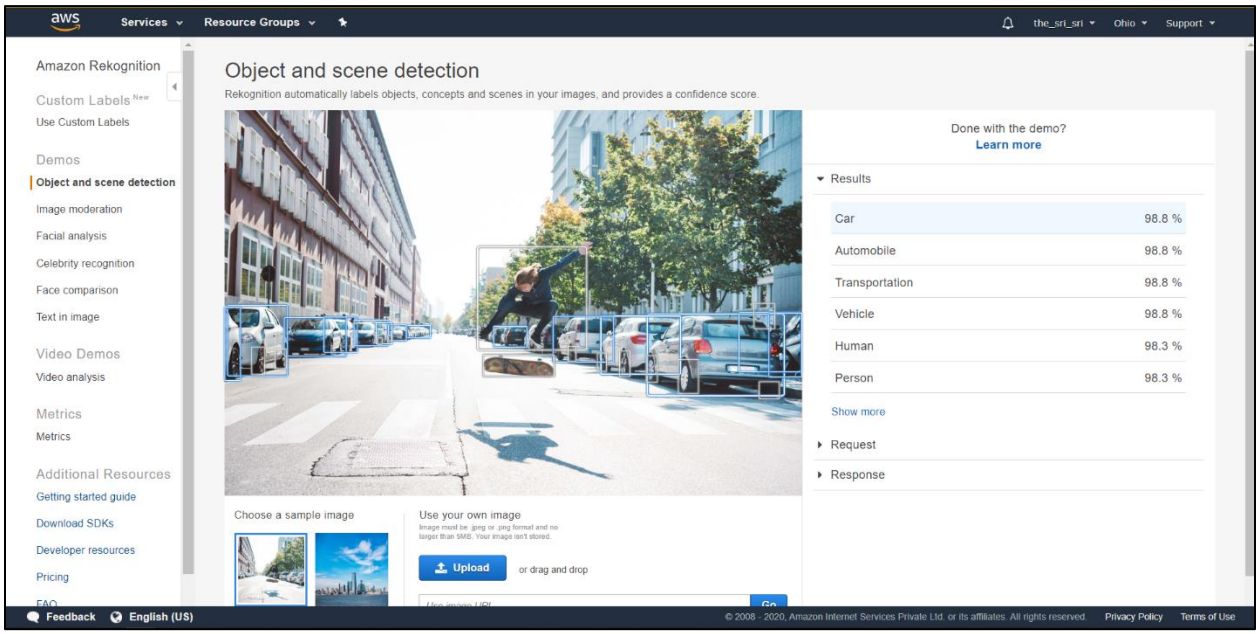
```
ec2-user@ip-172-31-44-120:/var/www/html/face
Redirecting to /bin/systemctl start httpd.service
[ec2-user@ip-172-31-44-120 ~]$ cd /var/www/html
[ec2-user@ip-172-31-44-120 html]$ cd face
[ec2-user@ip-172-31-44-120 face]$ sudo php -d memory_limit=-1 ~/composer.phar re
quire aws/aws-sdk-php
Using version ^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
b97ea33b5842c7894b804923c6c05580.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

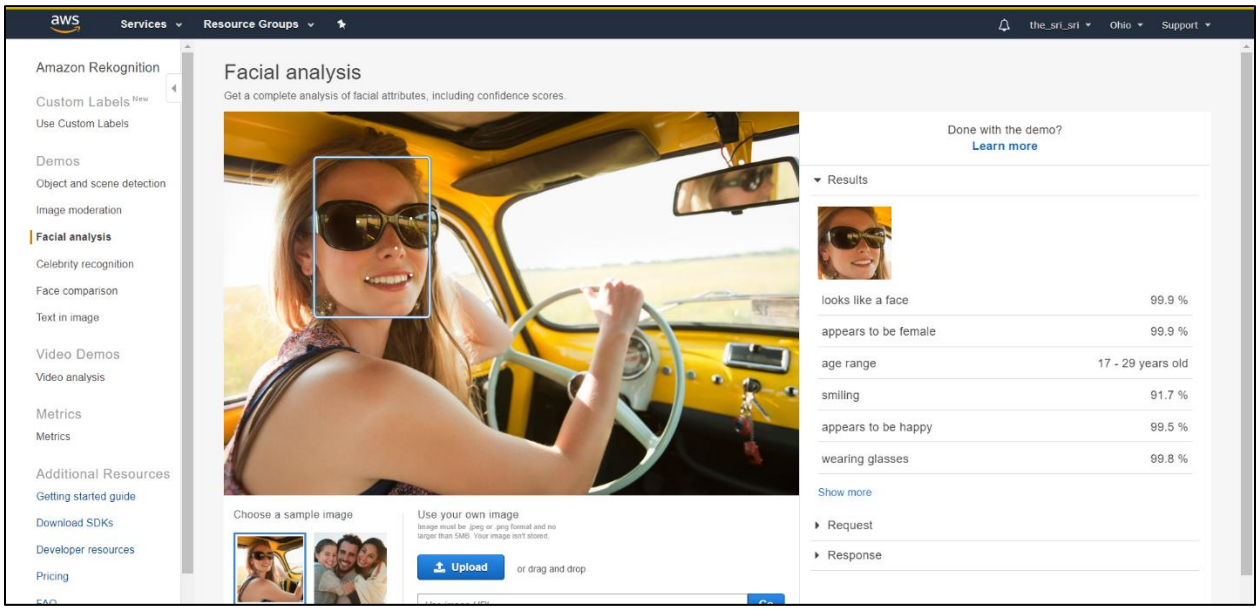


STEP 6: - A tour to **AWS rekognition.**

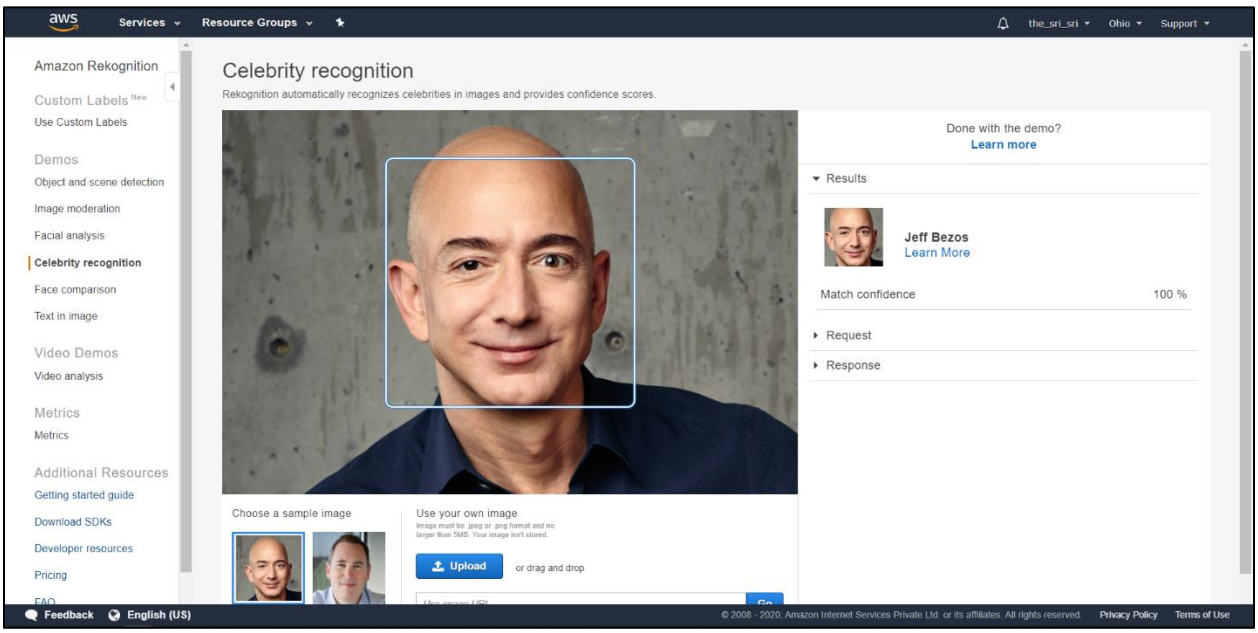
- ✓ **Object and scene detection**



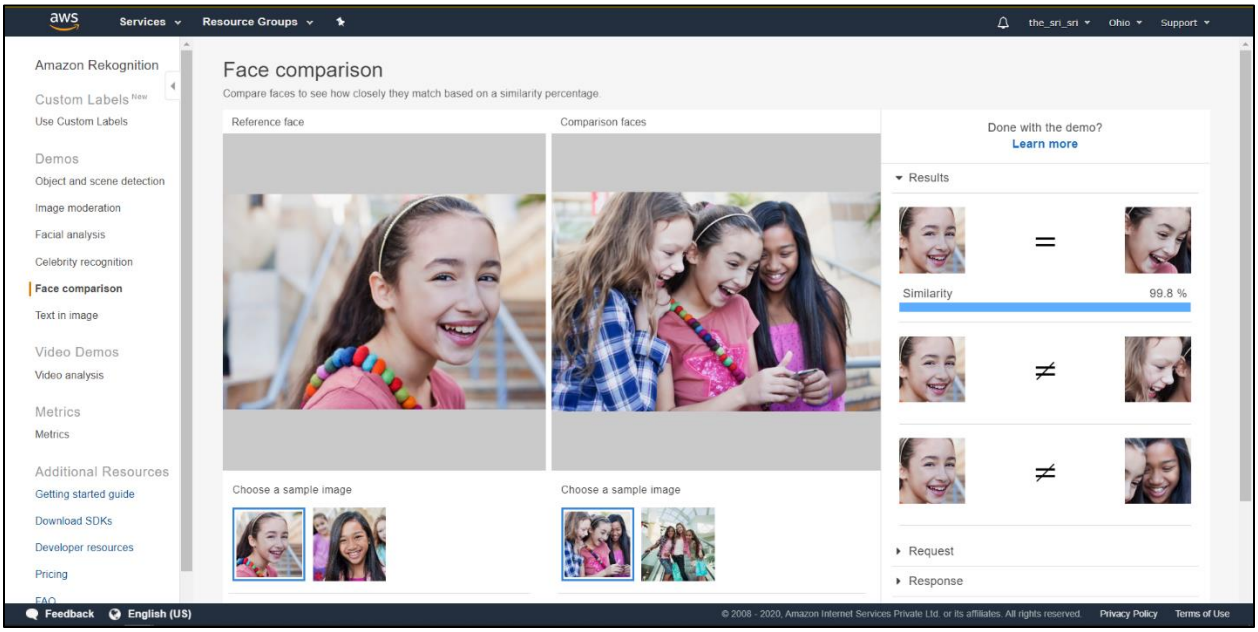
- ✓ **Facial Analysis**



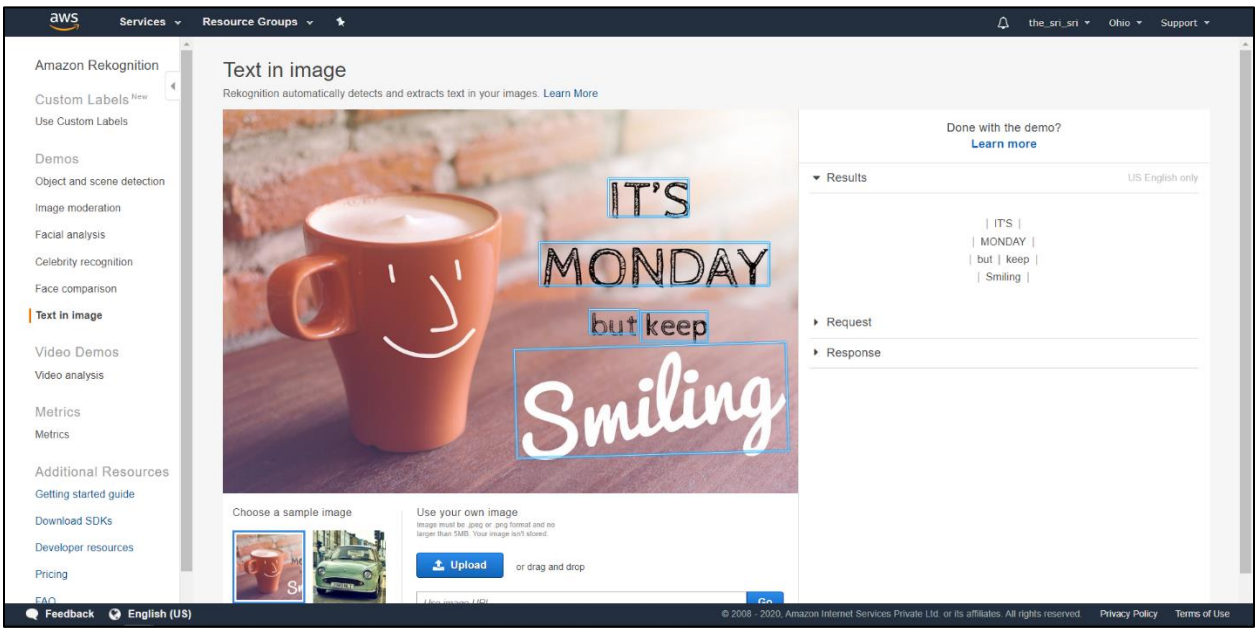
✓ Celebrity Recognition



✓ Face comparison



✓ **Text in image**



STEP 7: Using **AWS Rekognition** .

input Image :-

Output :-

```
ec2-user@ip-172-31-44-120:/var/www/html/face
login as: ec2-user
Authenticating with public key "imported-openssh-key"
Last login: Mon Mar 30 13:14:11 2020 from 106.208.181.95

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

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-44-120 ~]$ cd /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]$ 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 vim index.php
[ec2-user@ip-172-31-44-120 face]$ sudo vim index.php
[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/team.jpg[ec2-user@ip-172-31-44-120 face]$
```