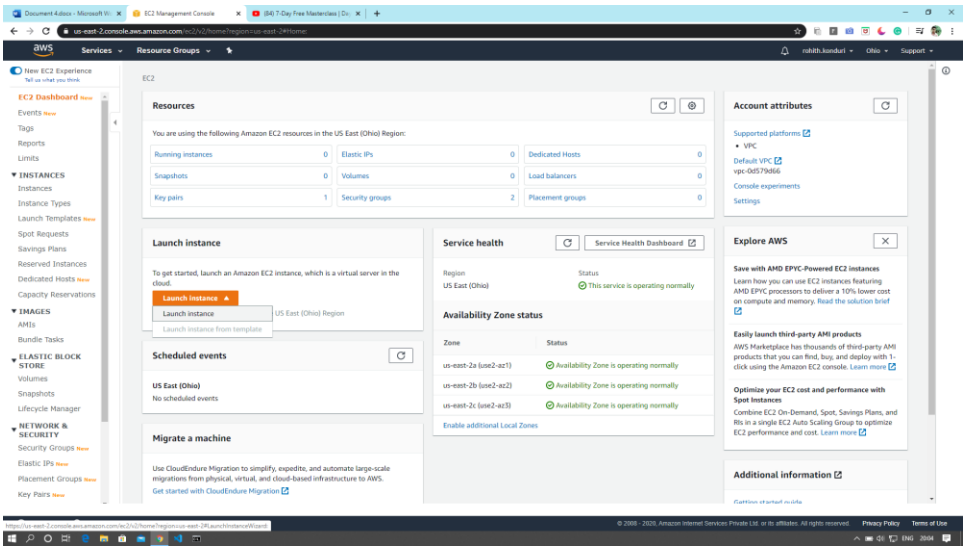
DAY 01

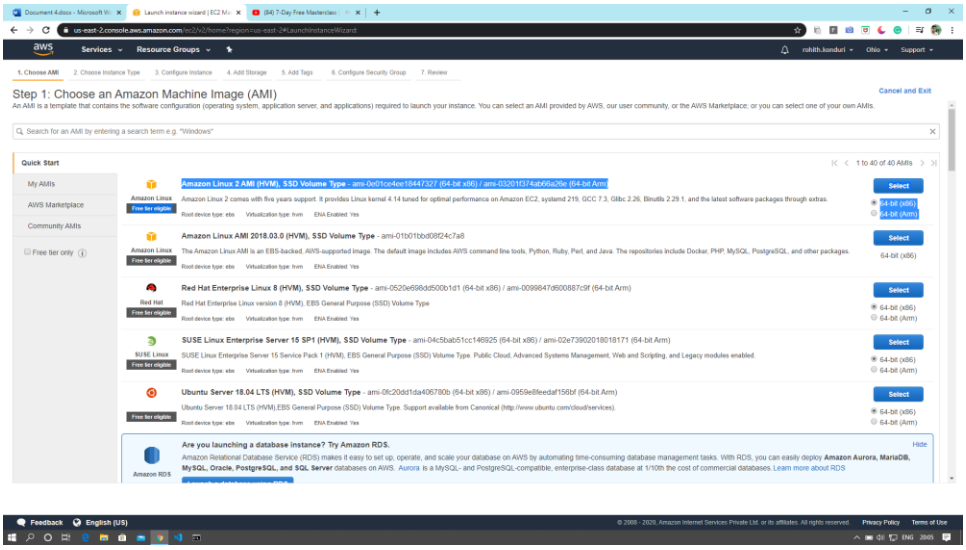
1)Created AWS account.

2)Creating and launching an Instance by using EC2 service.

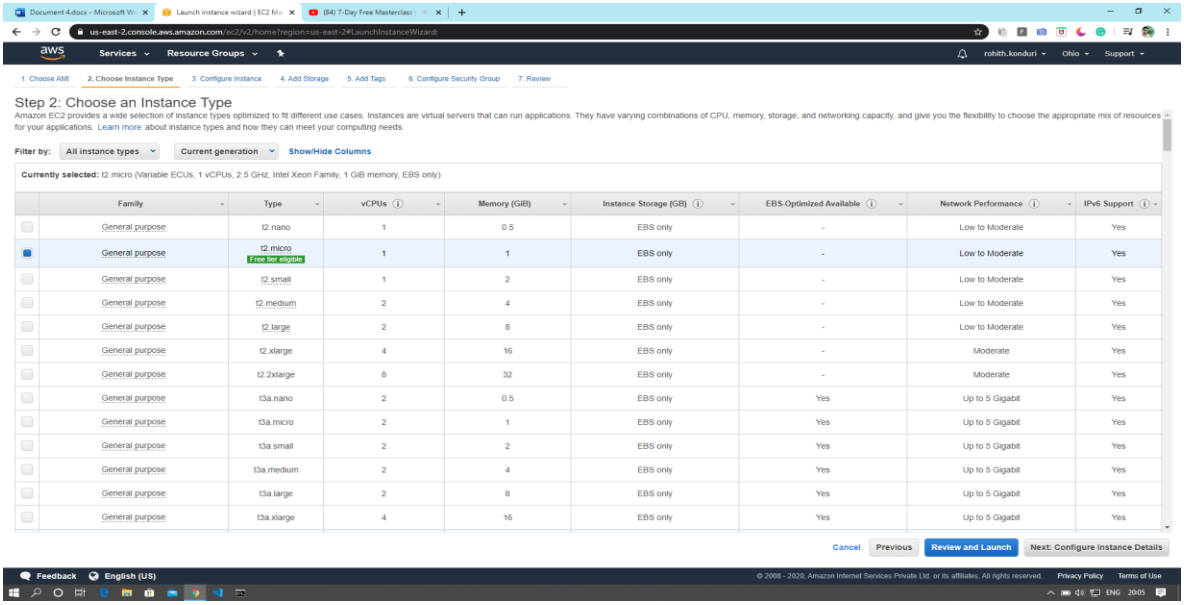
Launching instance



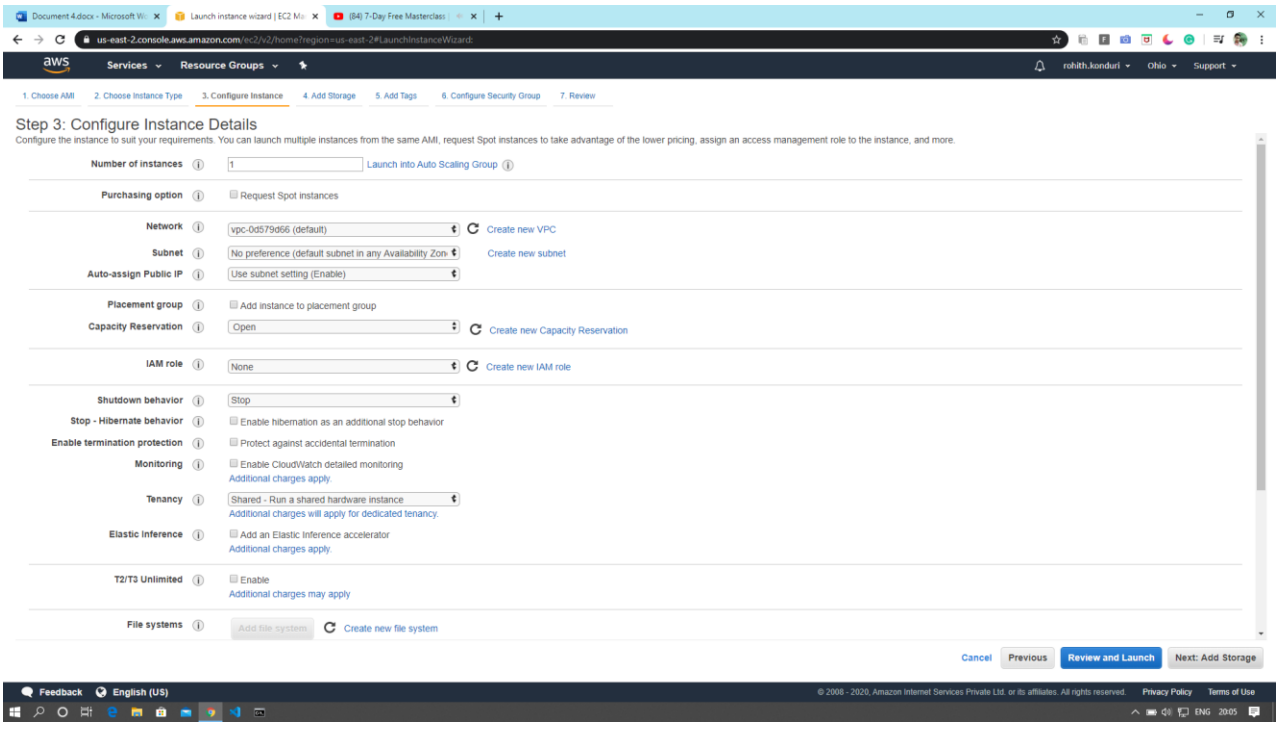
Step 1: Choosing AMI



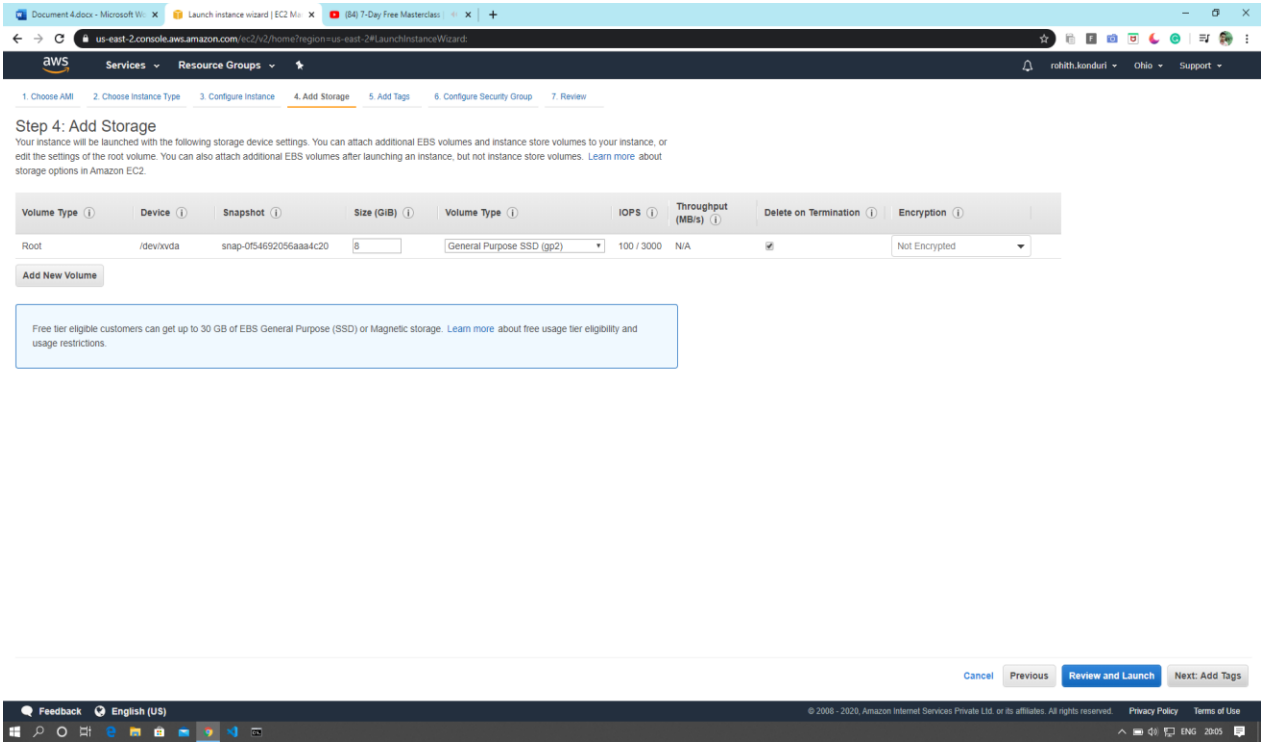
Step 2: Choosing Instance type (Selecting t2.micro which is free)



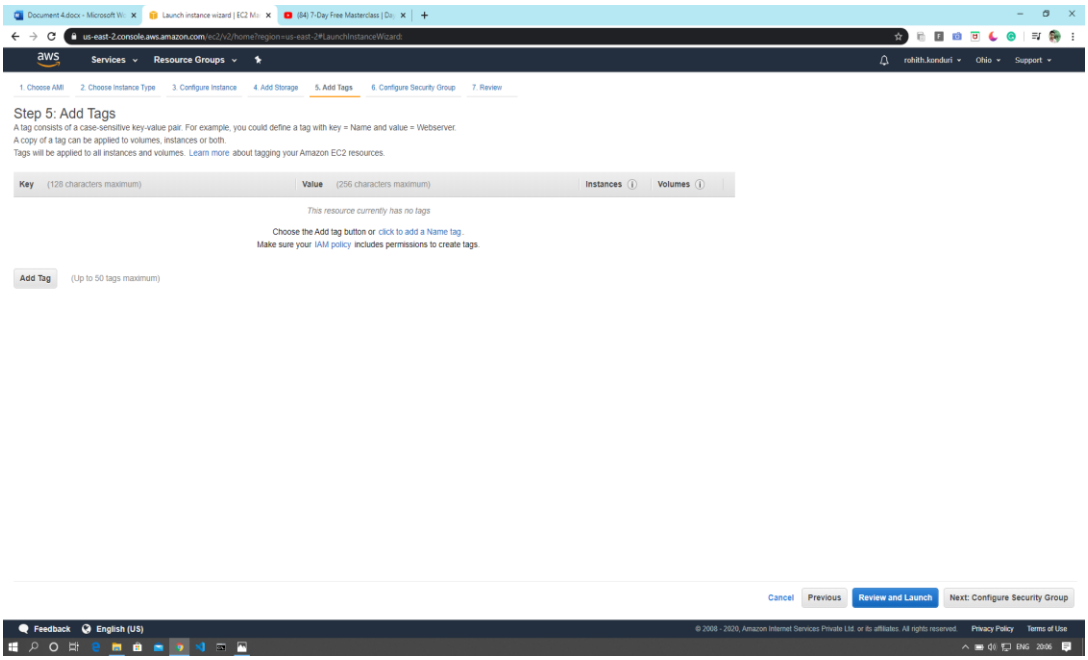
Step 3: No change in step 3



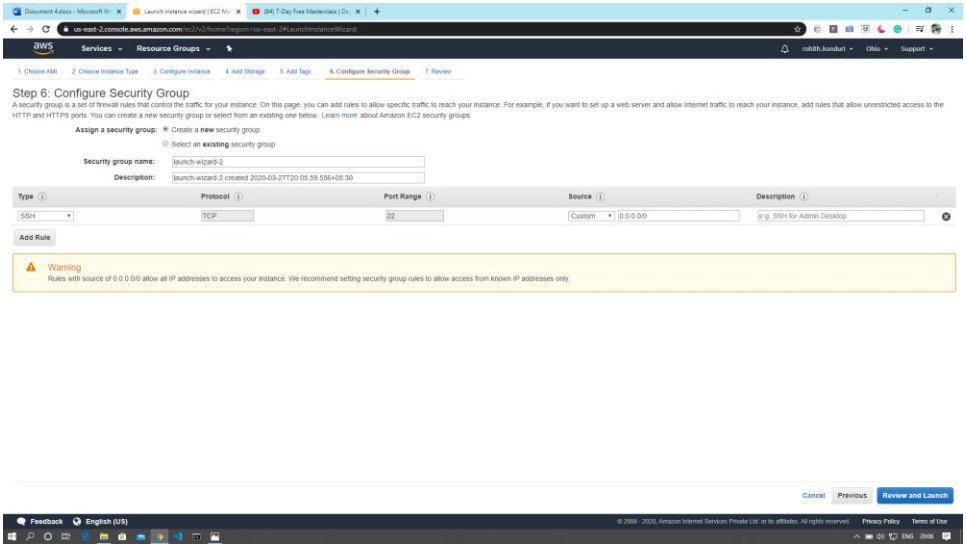
Step 4: Adding Storage (8GB)



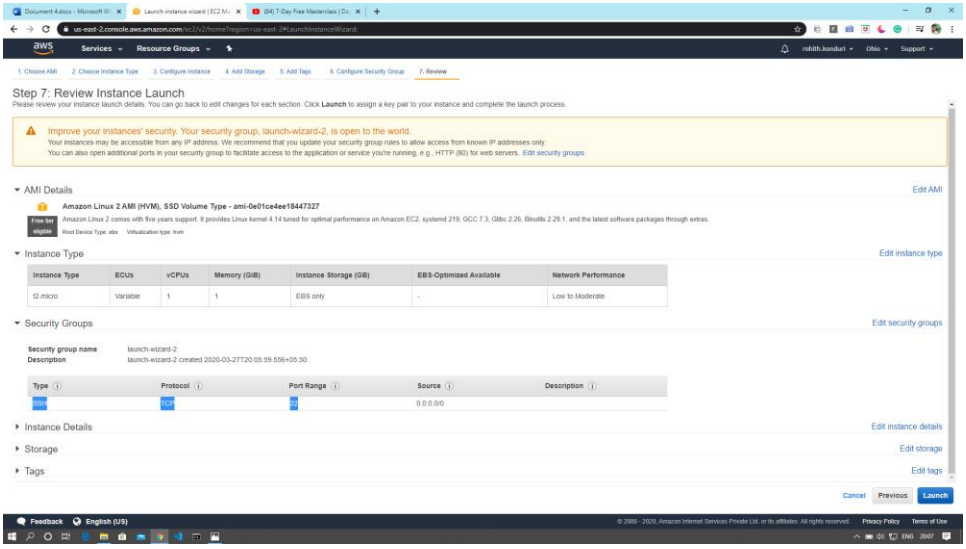
Step 5: No change in this step



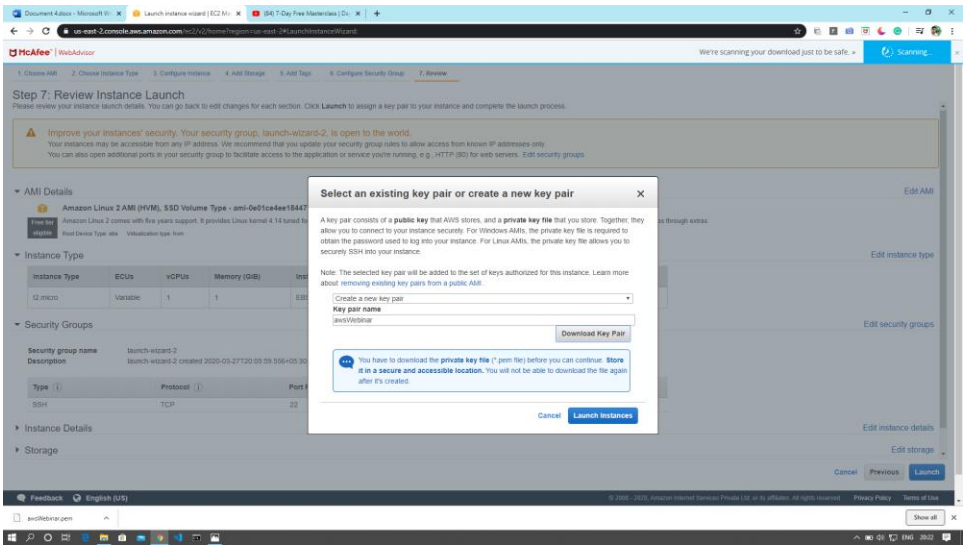
Step 6: Configuring security group (SSH type and port range =22)



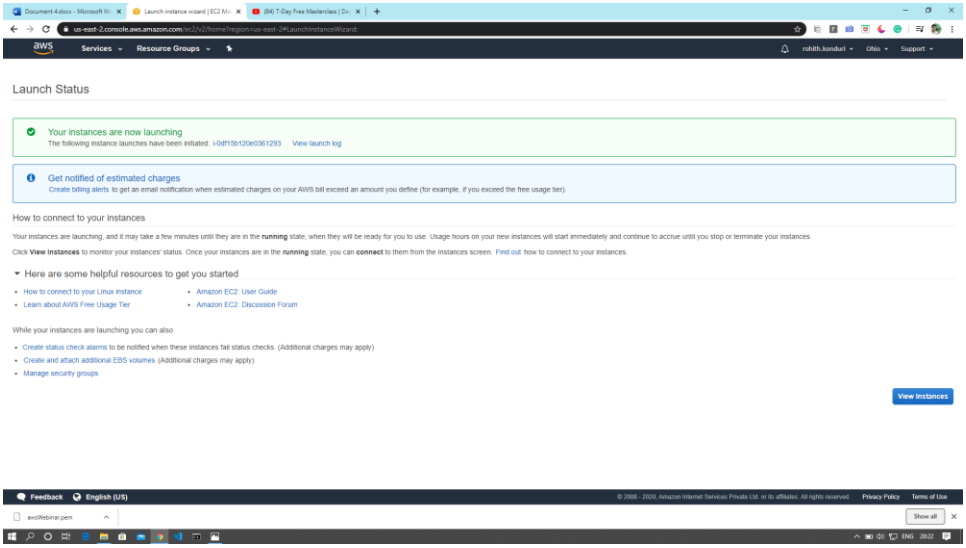
Step 7: Reviewing previous steps and launching



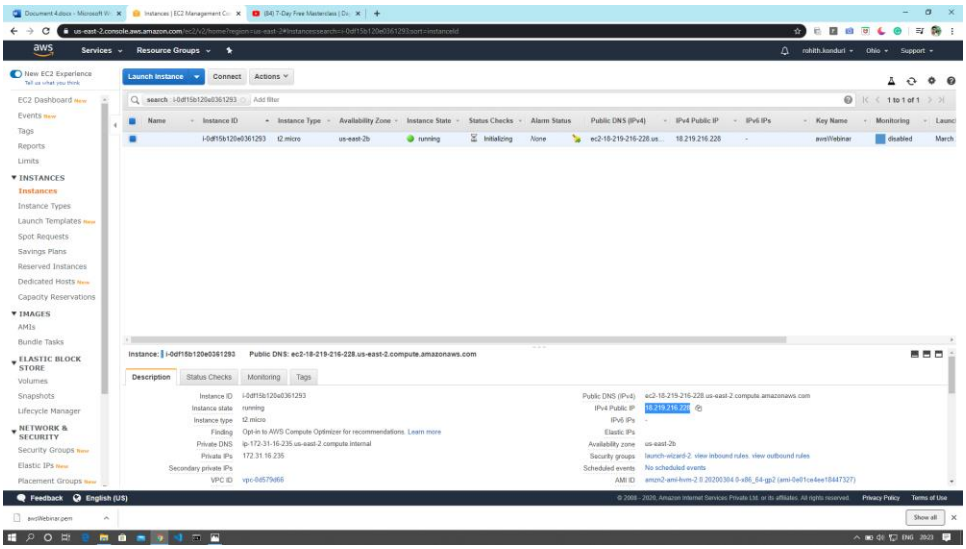
Select create a new key pair and name it.



This message will be displayed if your instance launch is successful.

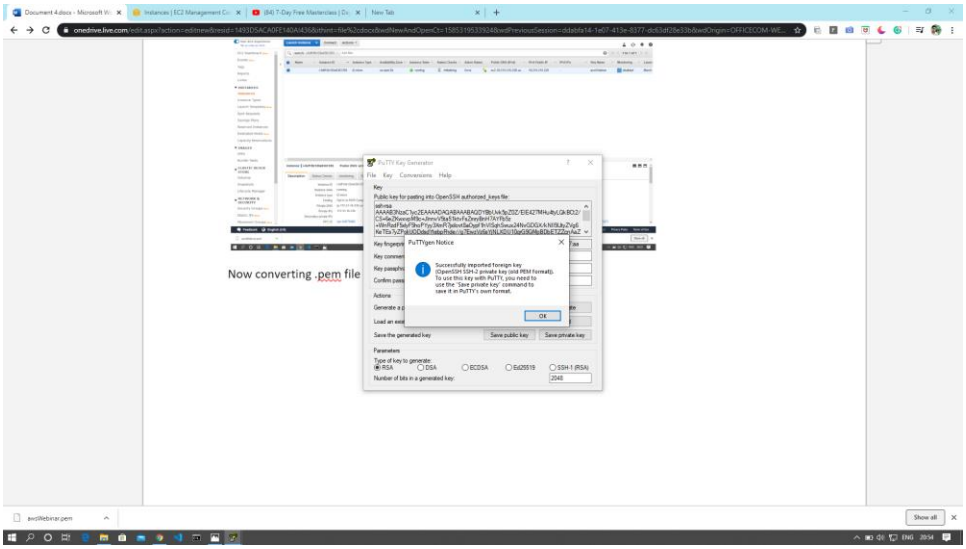


This is our instance.



Now converting .pem file to .ppk file.

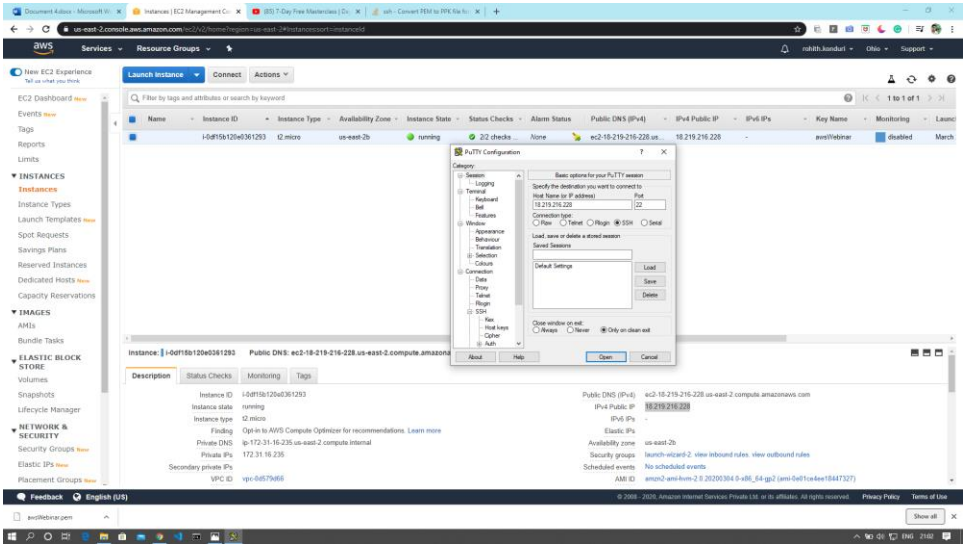
Load your .pem file in puttygen and save it as .ppk file by clicking save as private key.



.ppk file is ready!

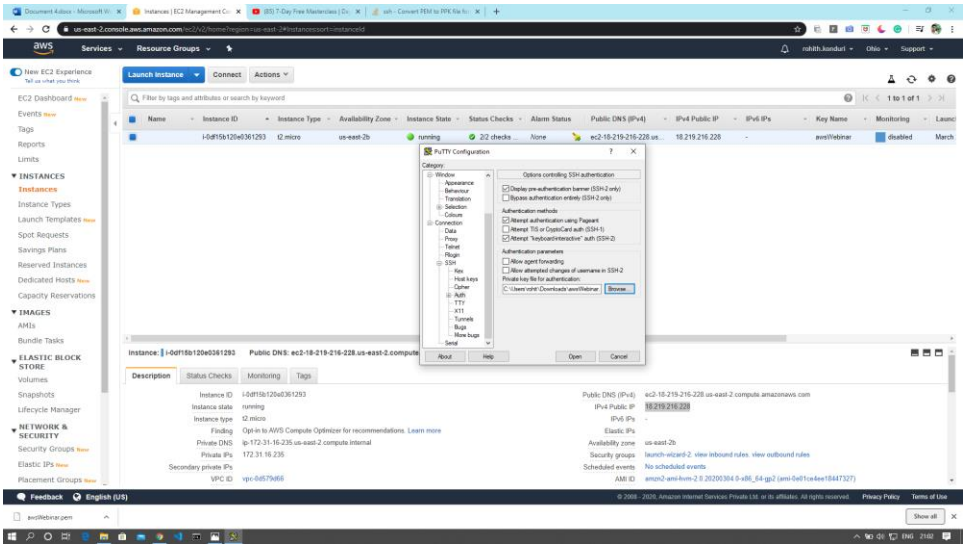
Now open putty and follow the given steps :

Enter ip address of our instance.

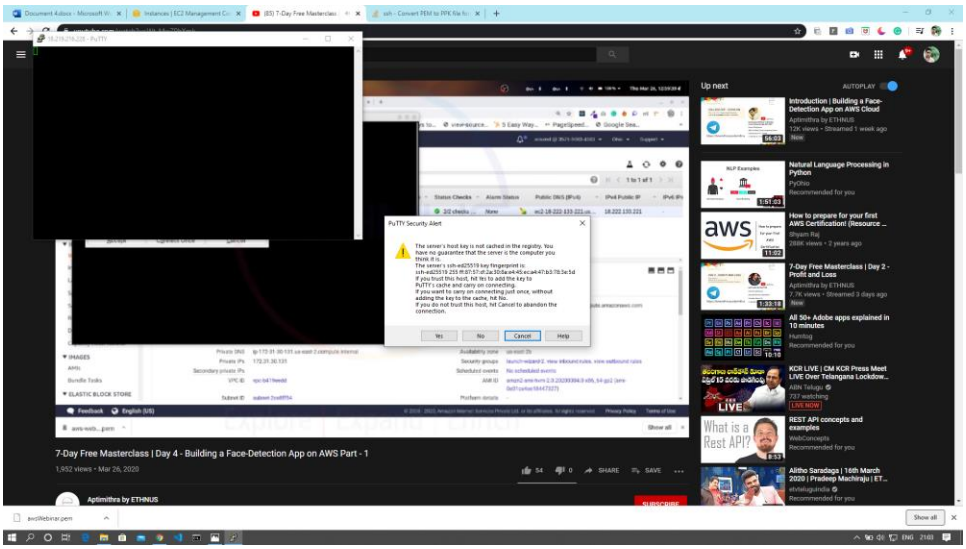


Go to SSH option and click on Auth option without opening it.

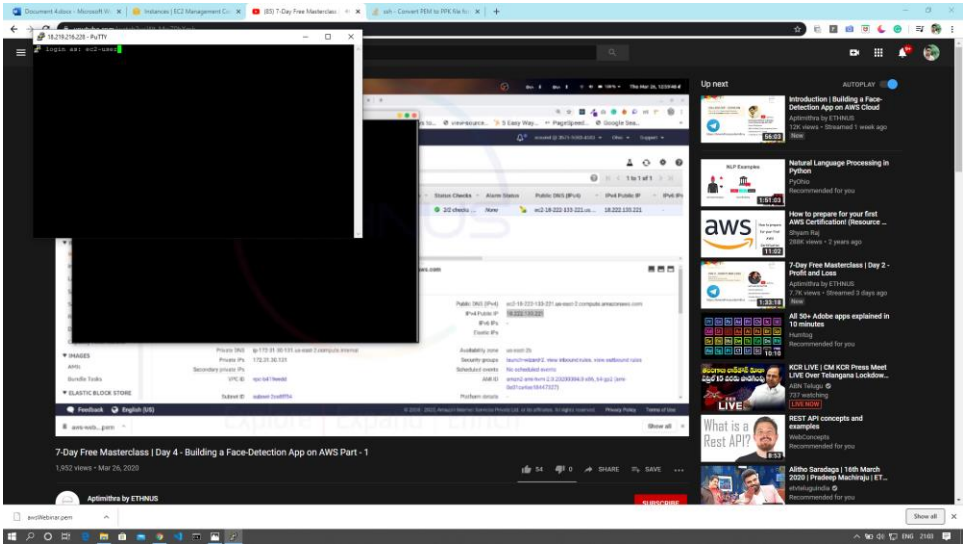
Now browse the file and load it (.ppk file).



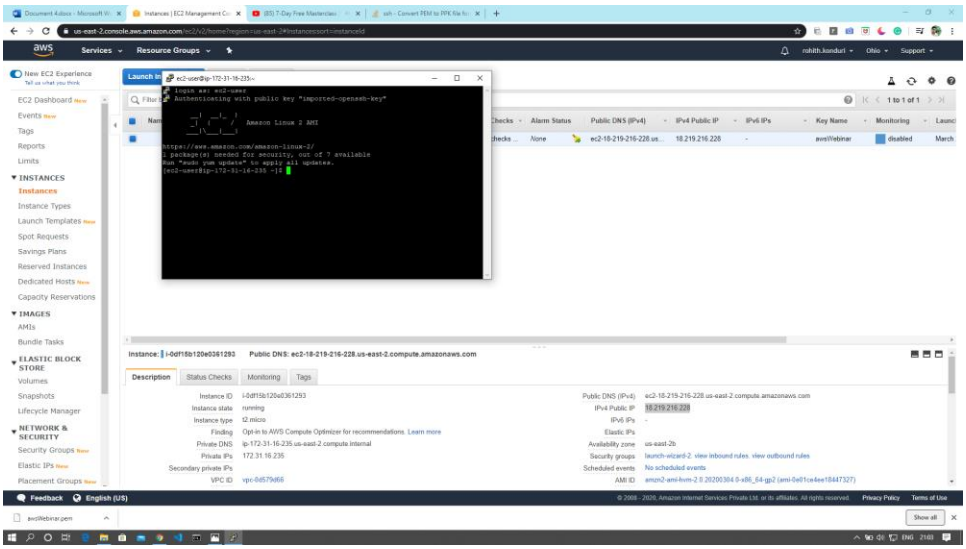
Now accept the message in the popup window



Now use your login name as “ec2-user”.

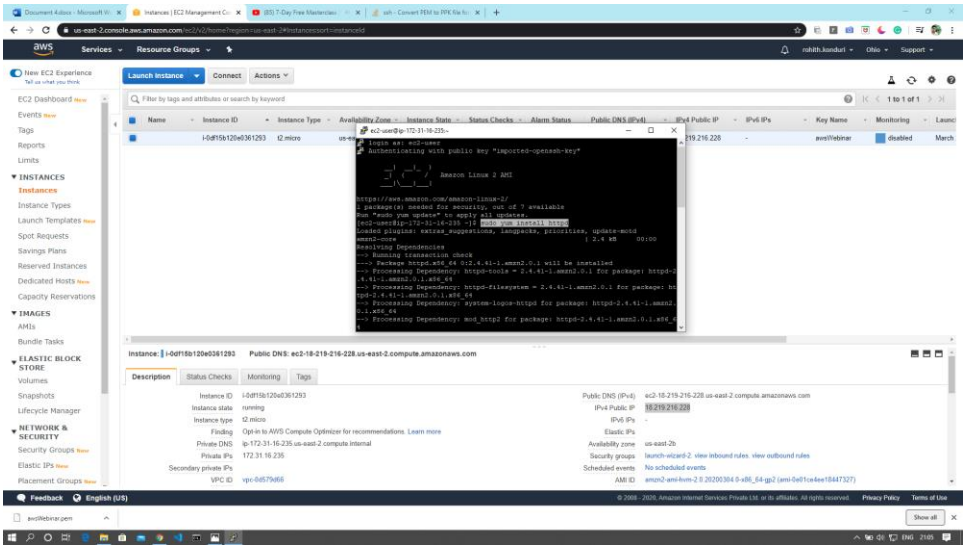


This will be displayed.



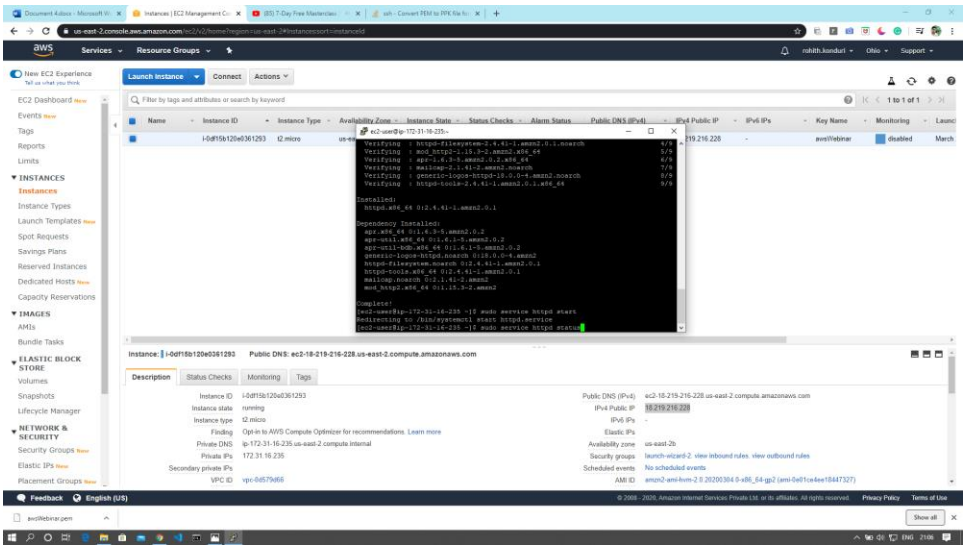
Now install httpd by using command – sudo yum install httpd

And click y to install it.

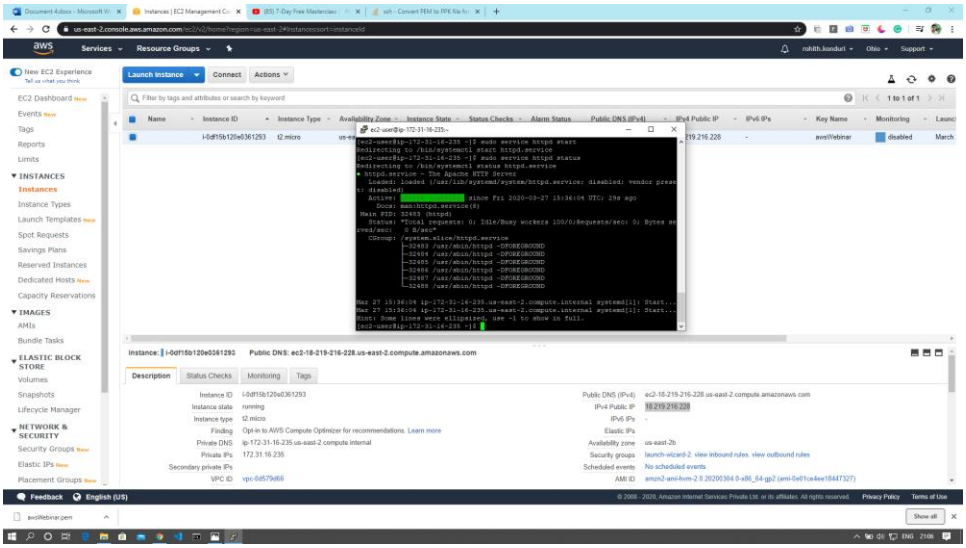


Use command – sudo service httpd start (to start the service)

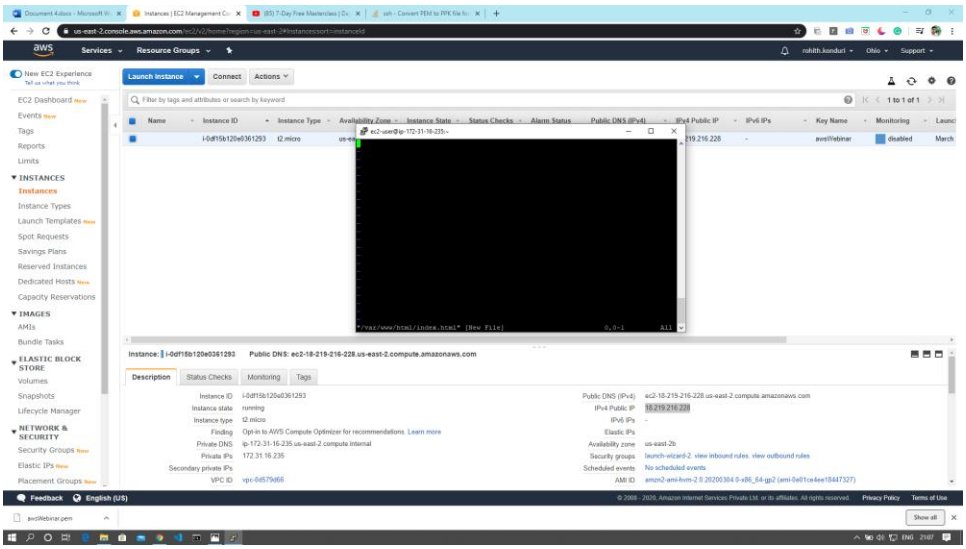
Use command – sudo service httpd status (to see status of the service)



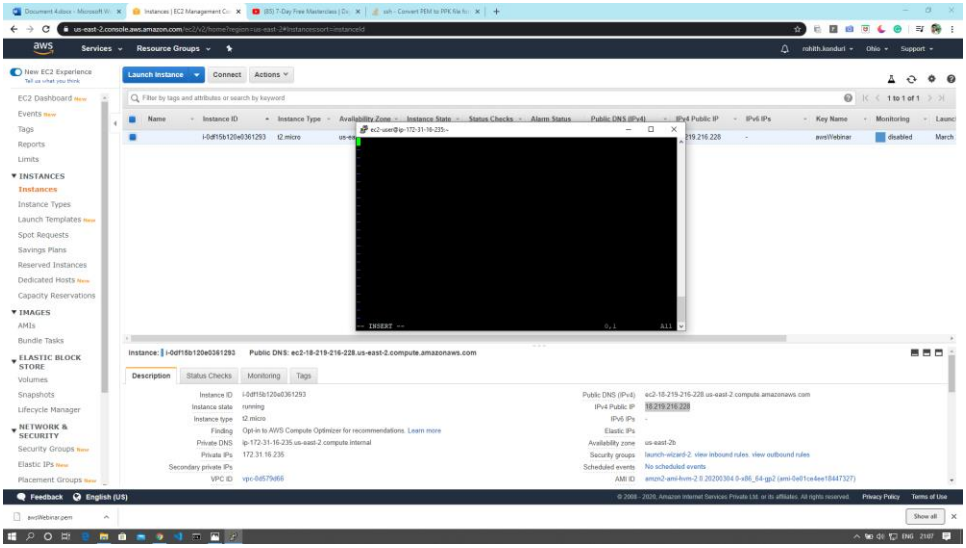
This green message will be displayed if everything is gone right!



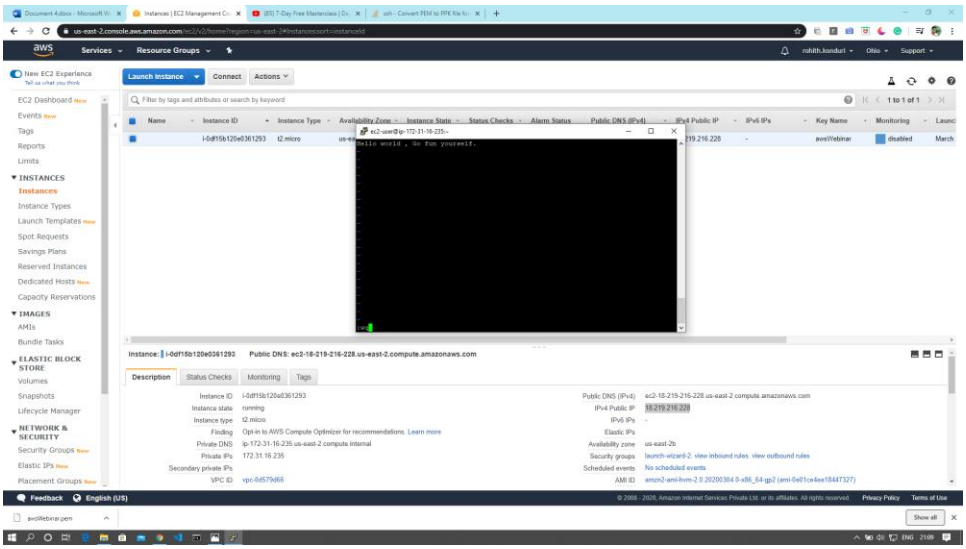
Now use command – vim /var/www/html/index.html



Press key ‘i’ to type whatever you want.



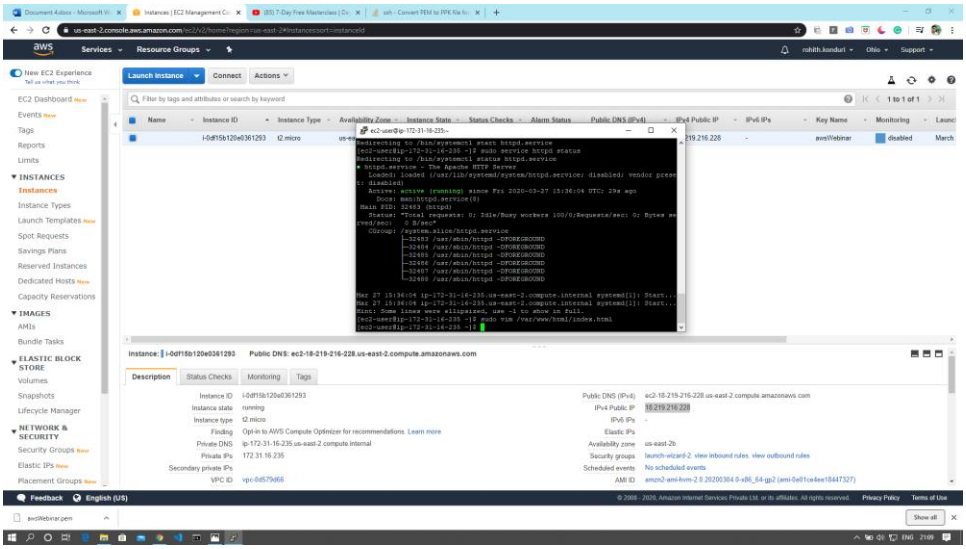
Then press esc and :wq to write and quit.



Now your ip address cannot be accessed.

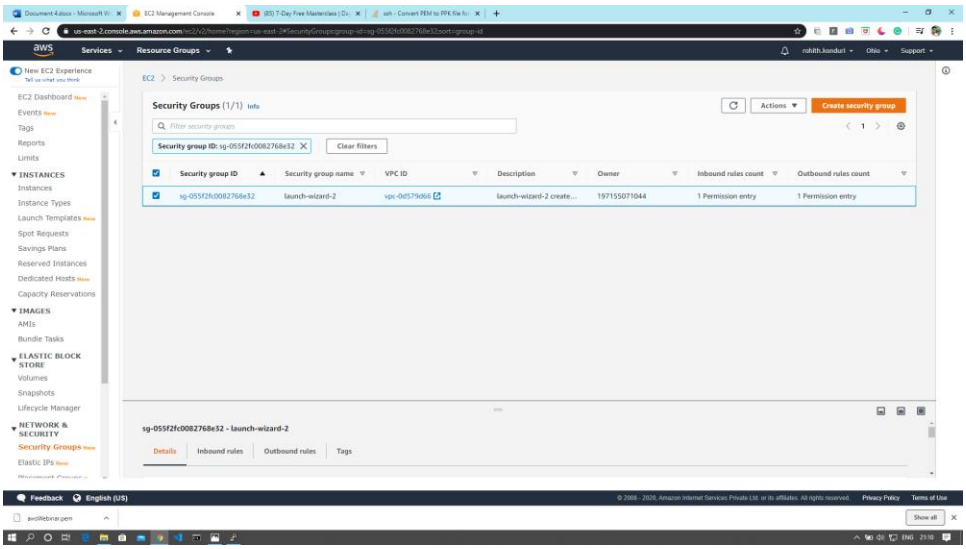
So we will change the security group inbound rules.

Go to security groups

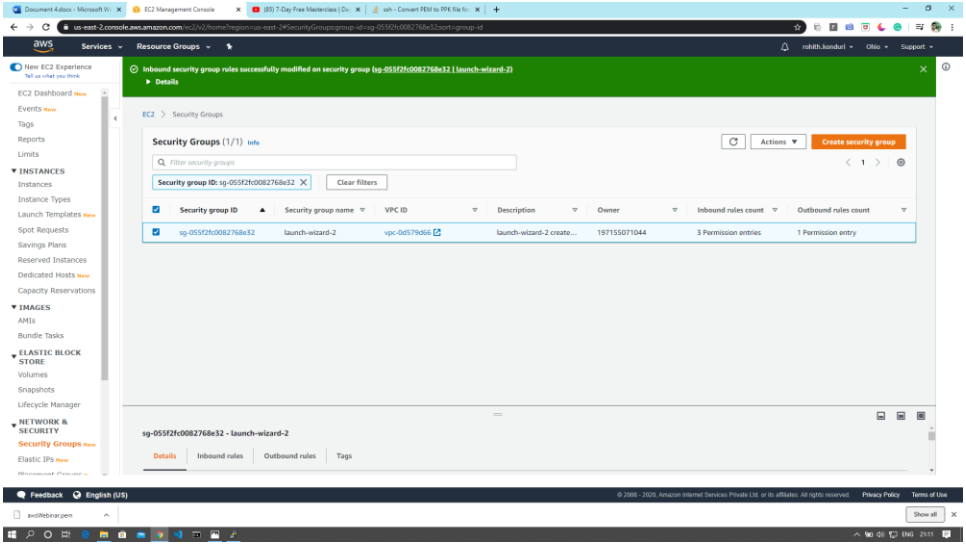


Click on edit inbound rules.

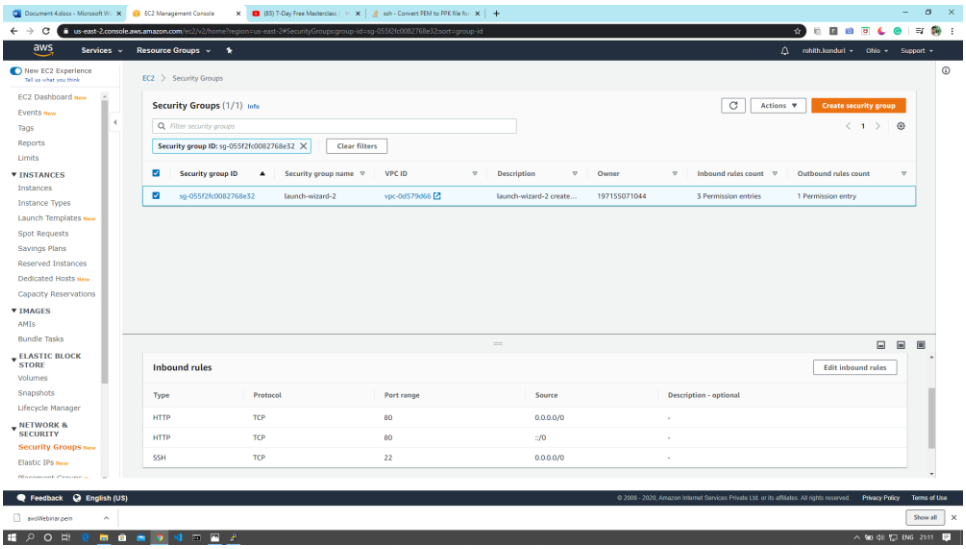
Add inbound rule http with port range 80 and change custom access to anywhere.



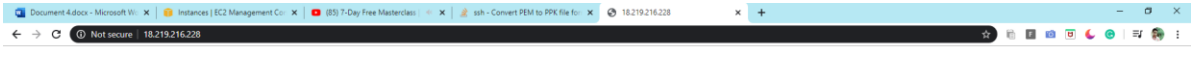
This message will be displayed.



Now your inbound rules will be changed to:



Now you can access your ip address from any where and the message you wrote previously will be displayed.



Hello world , Go fun yourself.

DAY 02

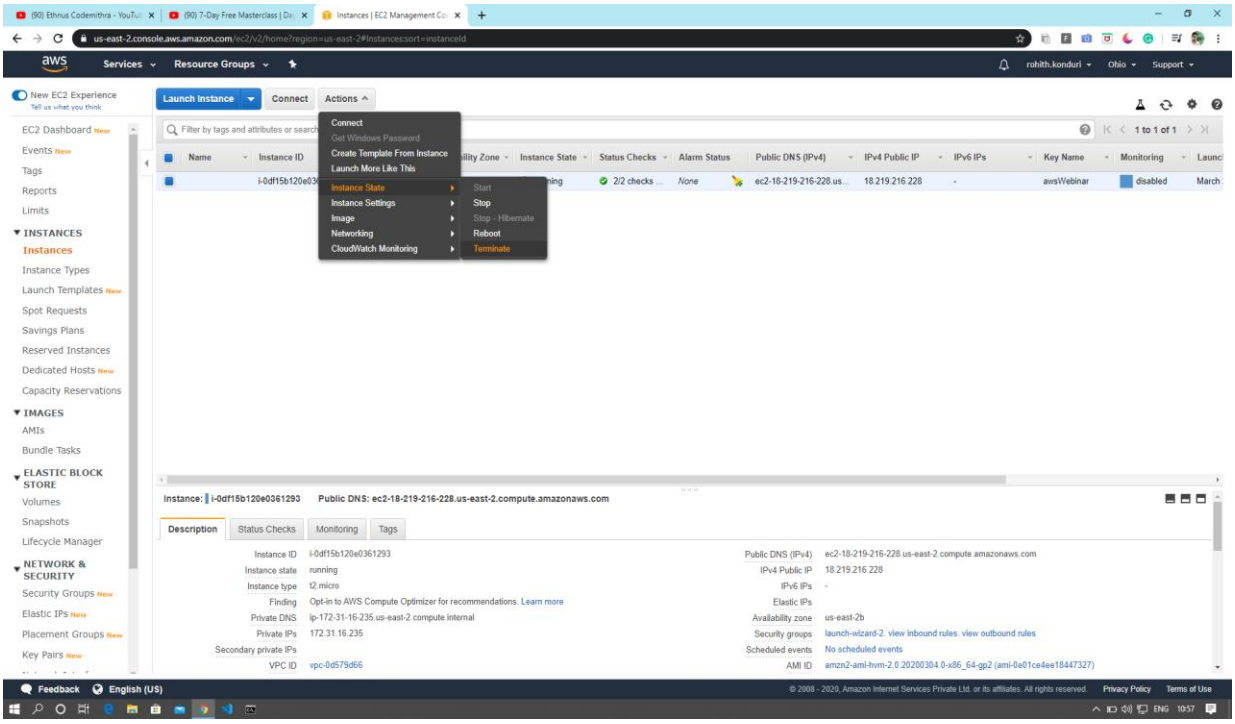
Changing instance state.

1)Stop. (You can again start it if you wish)

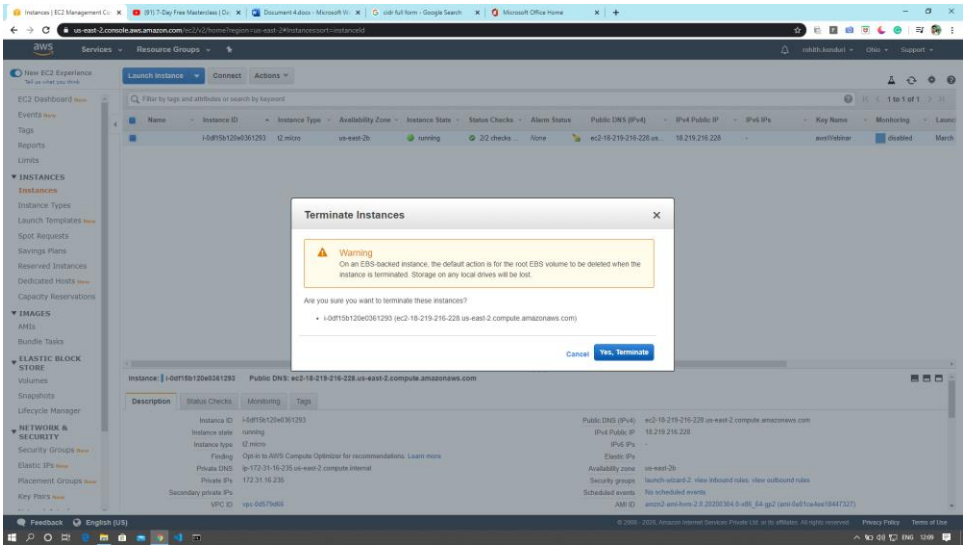
2)Reboot. (It’s like restarting your computer)

3)Terminate. (Once you terminate an instance, that’s it, it’s gone forever)

You can do those things here.

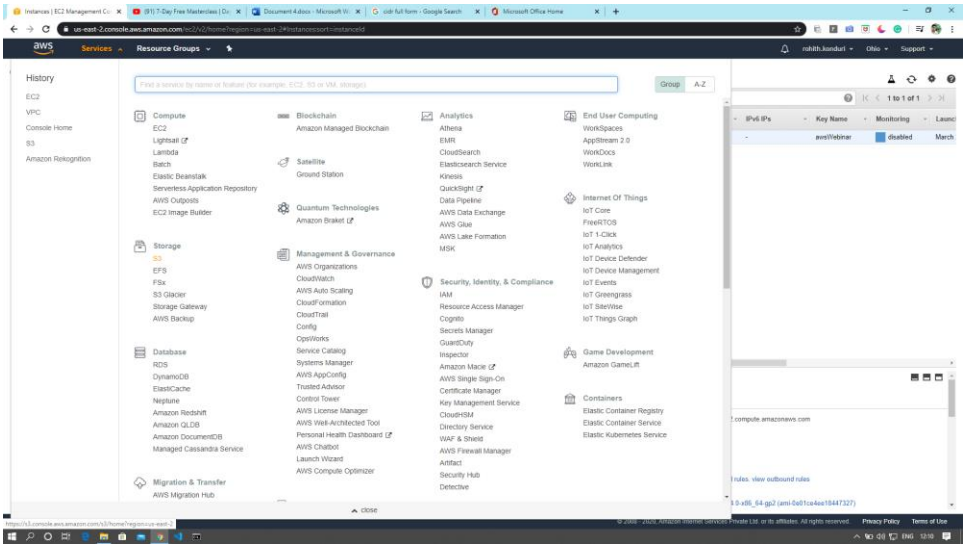


It will show a warning message, click on yes to terminate.



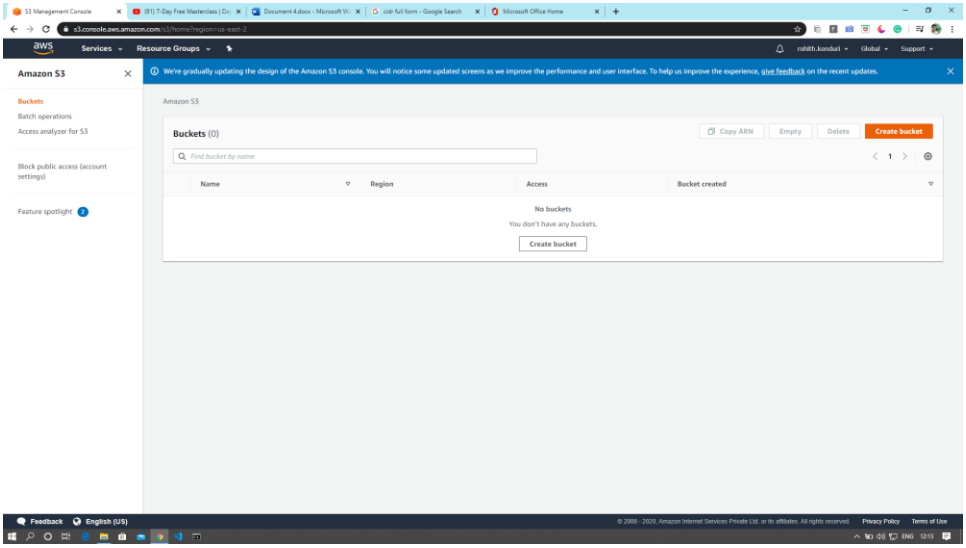
Now creating bucket in S3.

Go to S3 storage service in aws.

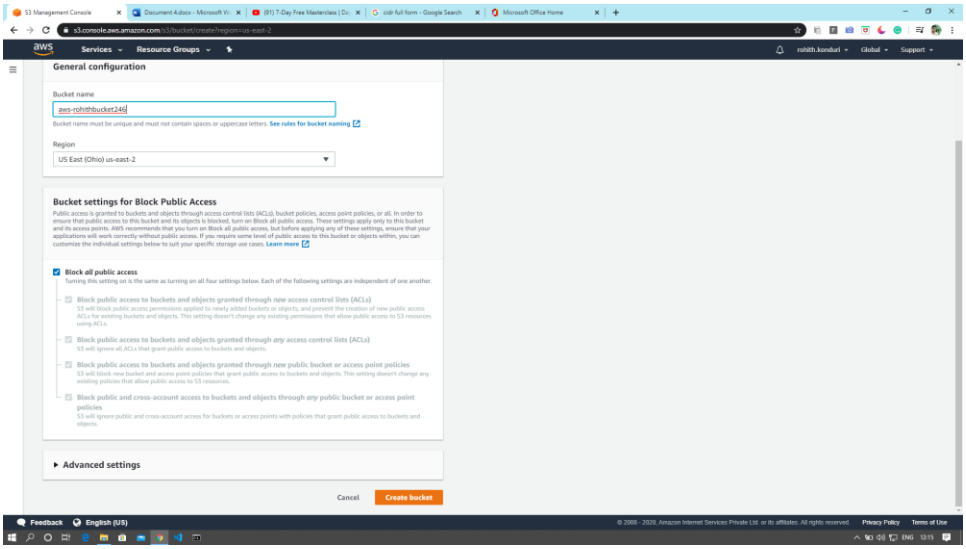


Then you will be redirected to S3 dashboard.

There click on create bucket to create one.

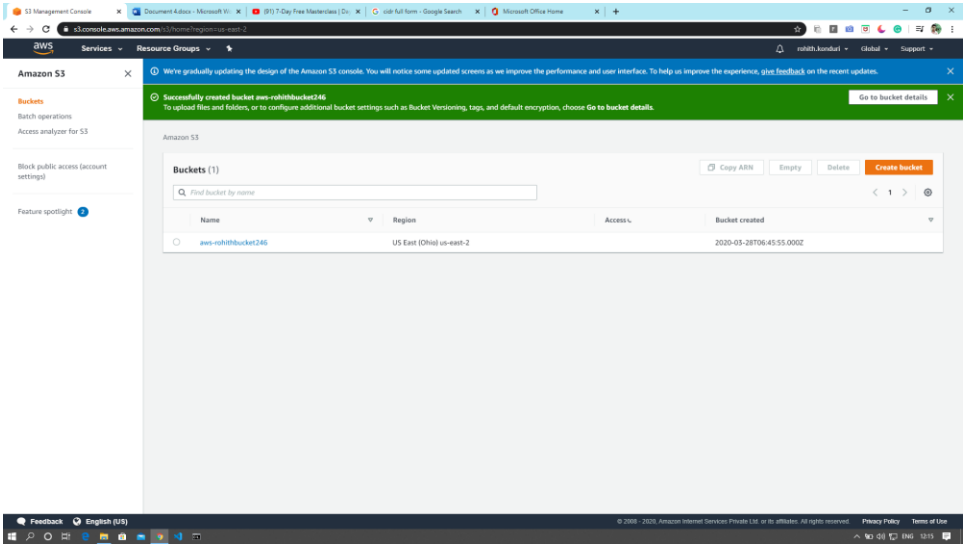


Create bucket with unique name.

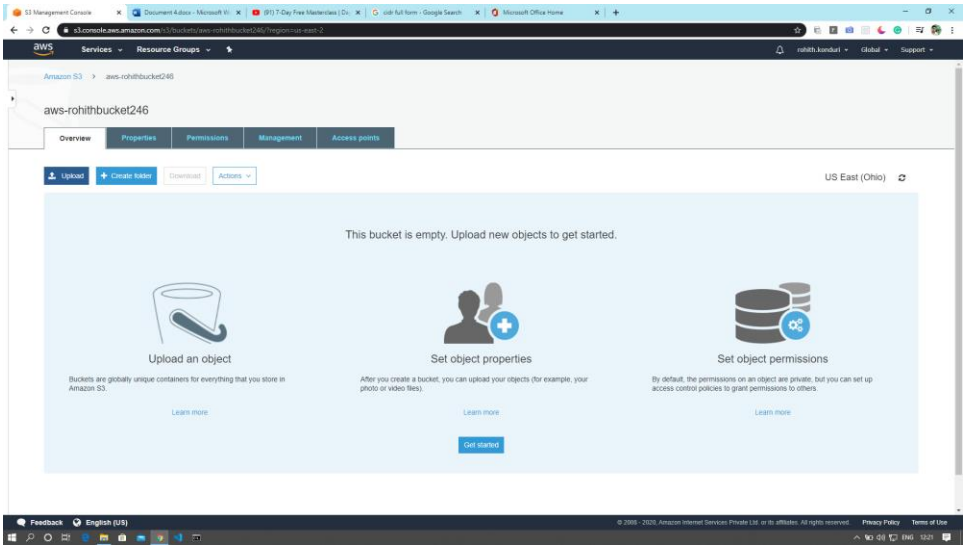


Click on create.

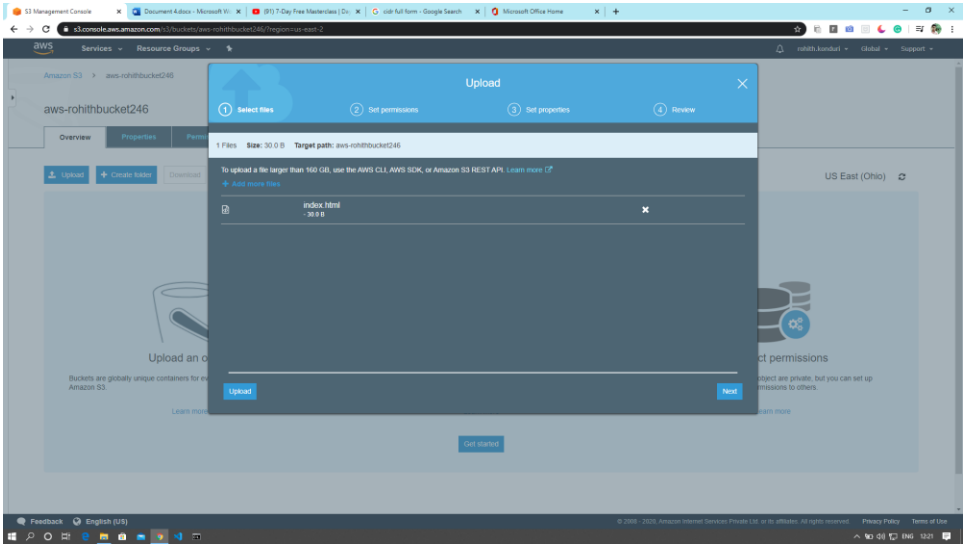
If your name is valid, then your bucket is created.



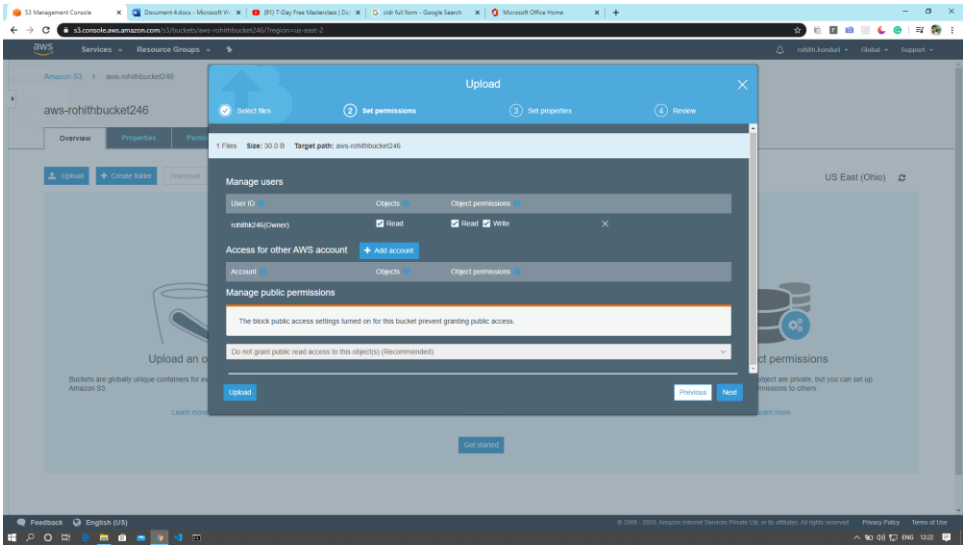
Now you can upload any object into this bucket.



Step 1: Upload your object



Step 2: Set permissions



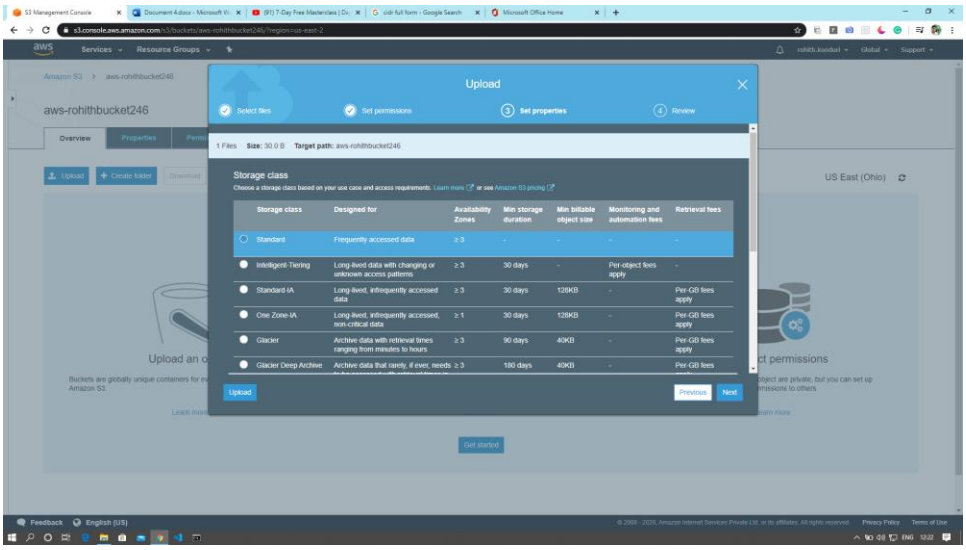
Step 3: Set properties

You can select standard (for your object to be in all availability regions)

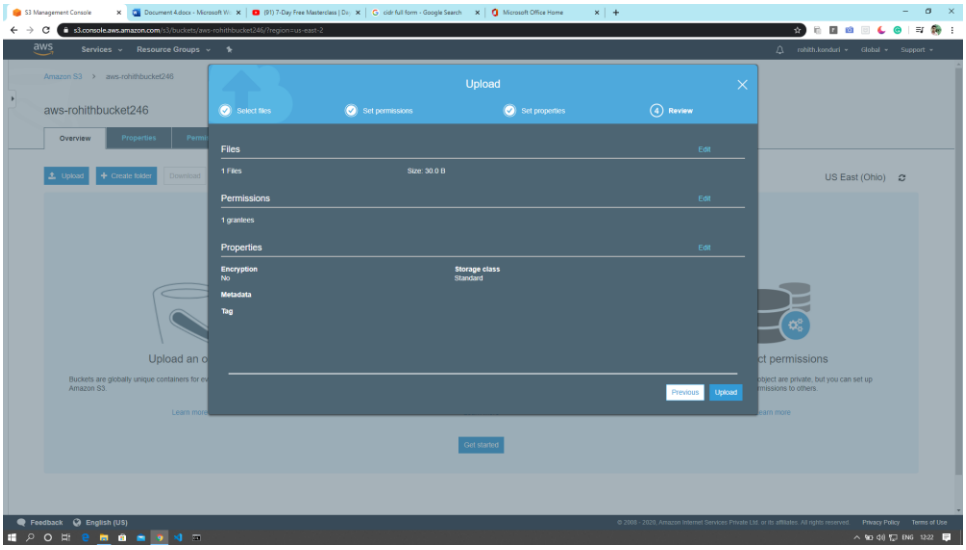
Or standard infrequent (if you access you object very less frequently)

Or one-zone infrequent (your oblect will be in only one of the availability regions)

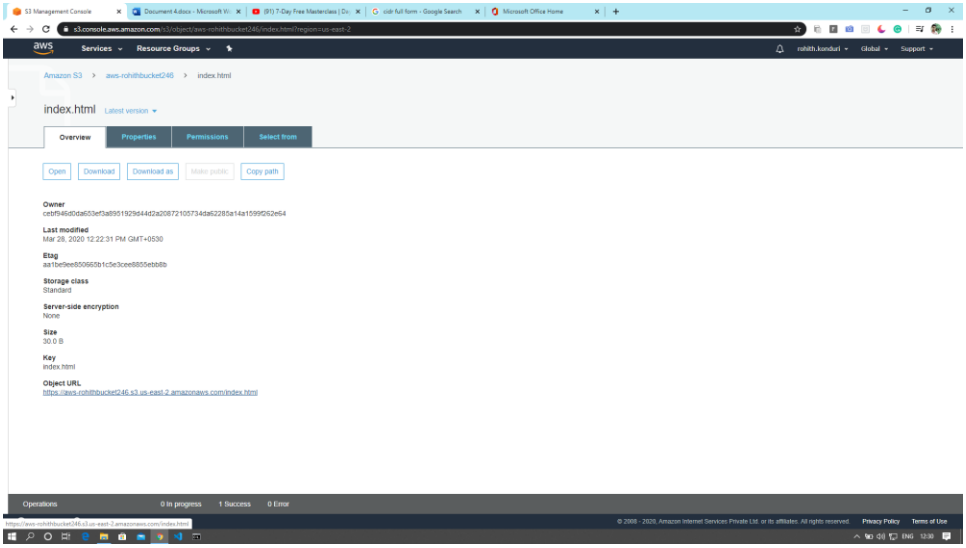
Or glacier (Like a cold storage)



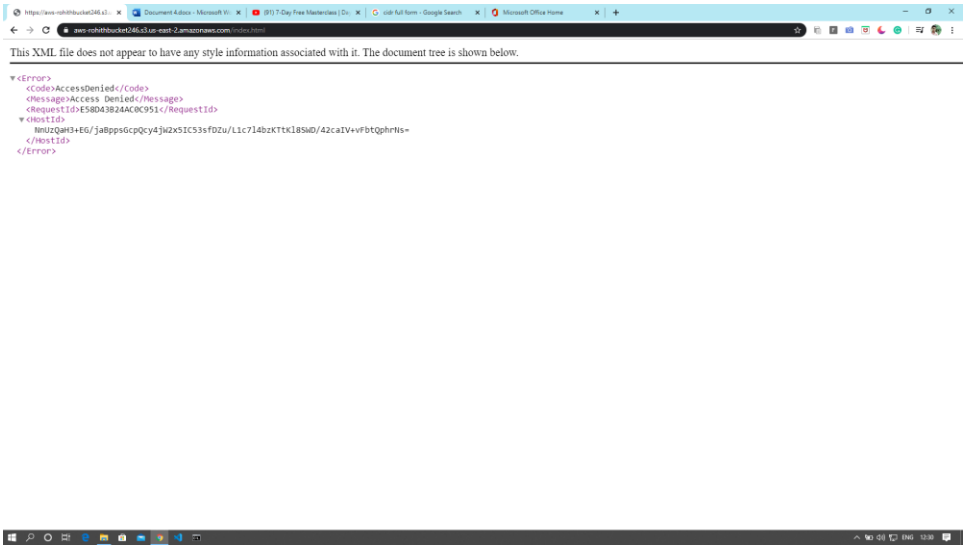
Step 4: Review option



Now you can access your object here.



Click on object url and this error will come

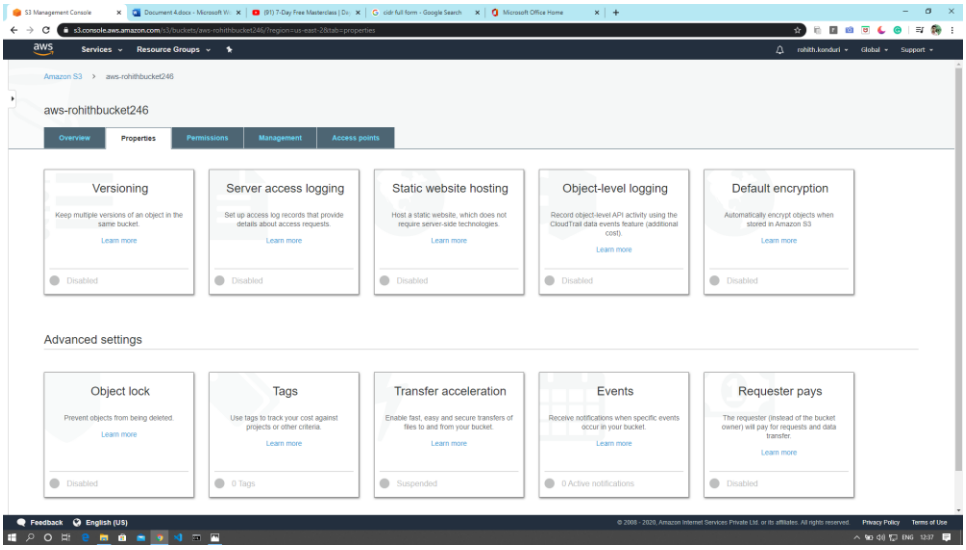


You can check amazon availability zones here,

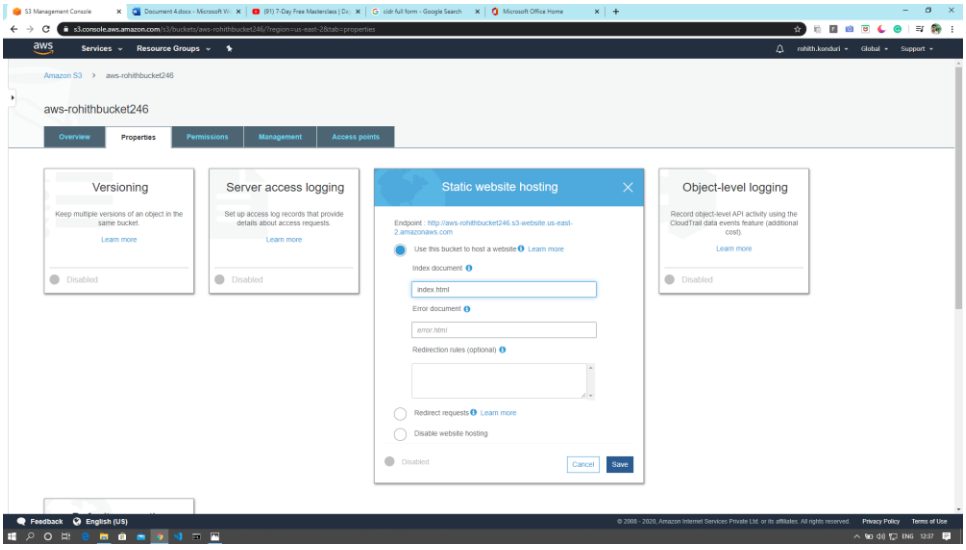
https://aws.amazon.com/about-aws/global-infrastructure/

The previous error can be cleared by static website hosting.

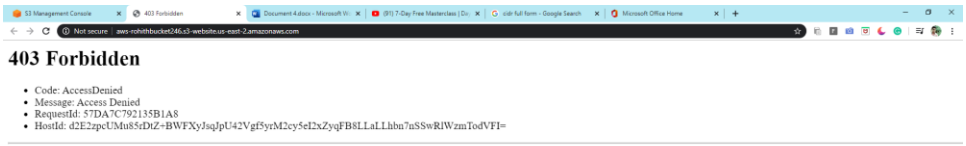
Go to properties of bucket.



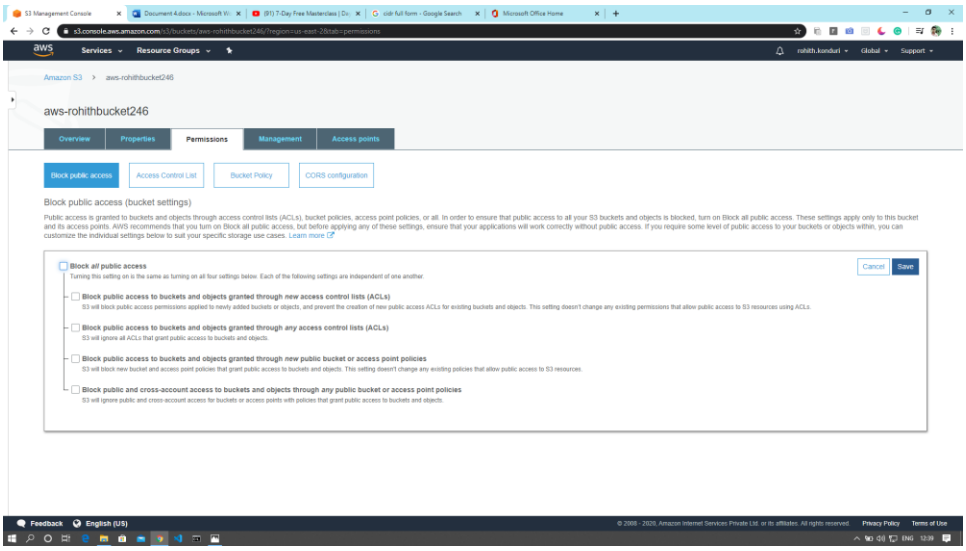
Put your object in static website hosting and save.



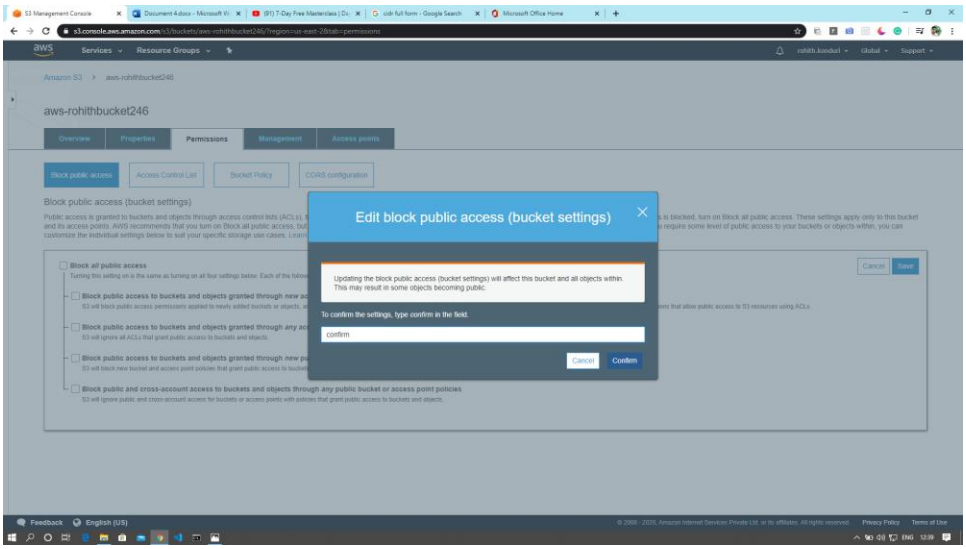
It will show this if you refresh your object url.



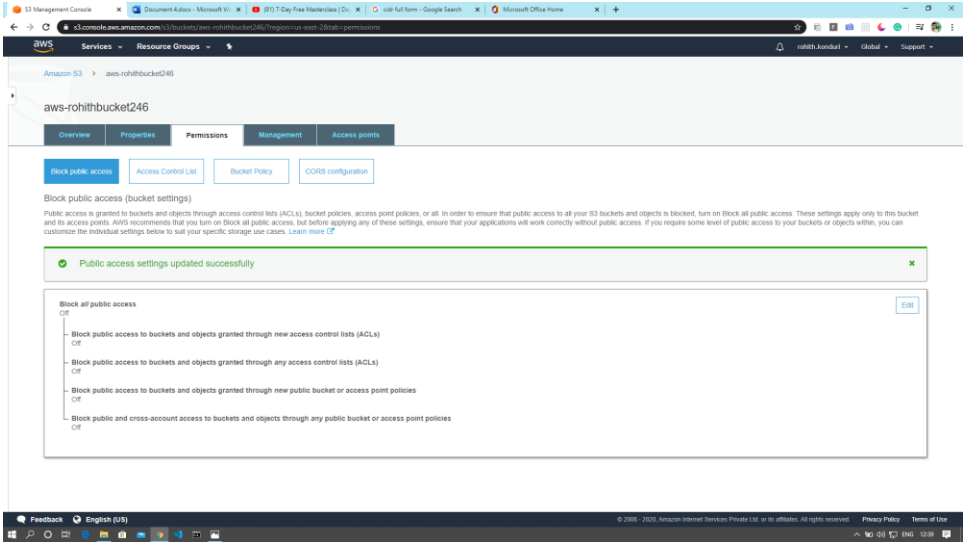
Now go to permissions of bucket and edit block public access and uncheck it.



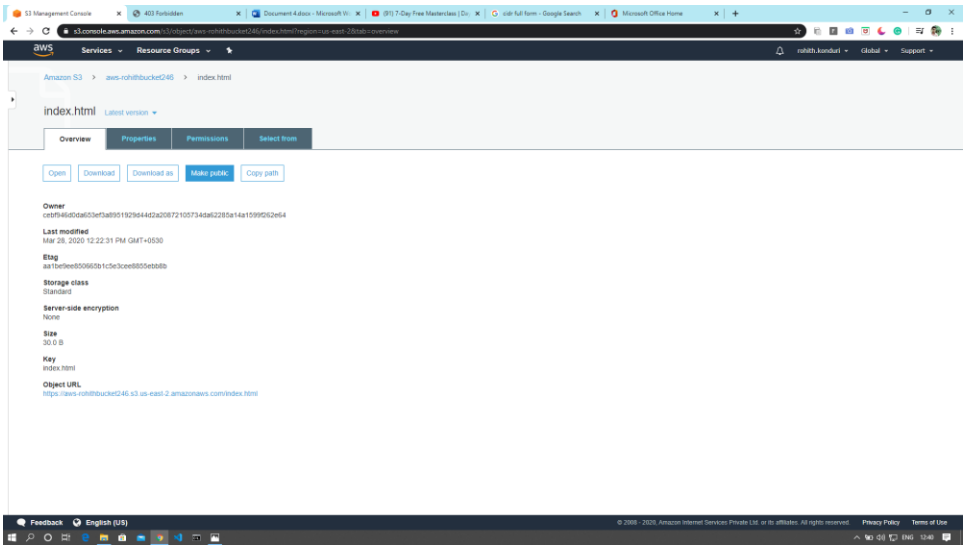
Type confirm in the pop up window and save



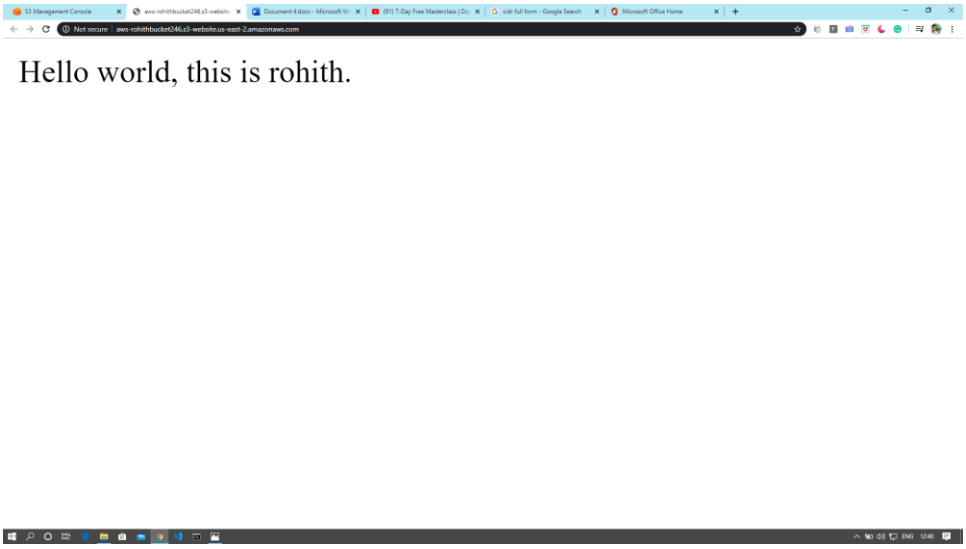
Public access will be updated.



Now go to object overview and click on make public.



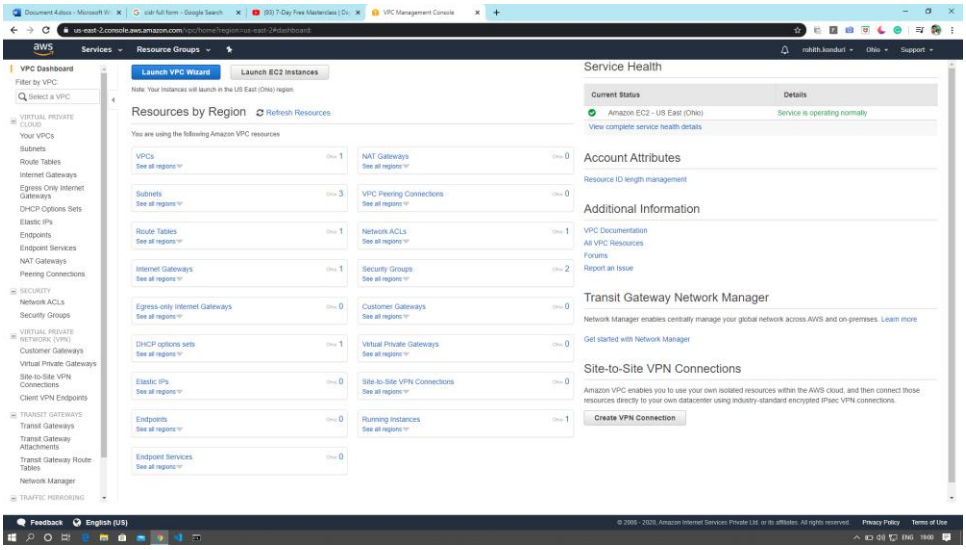
Now your object url will be working!



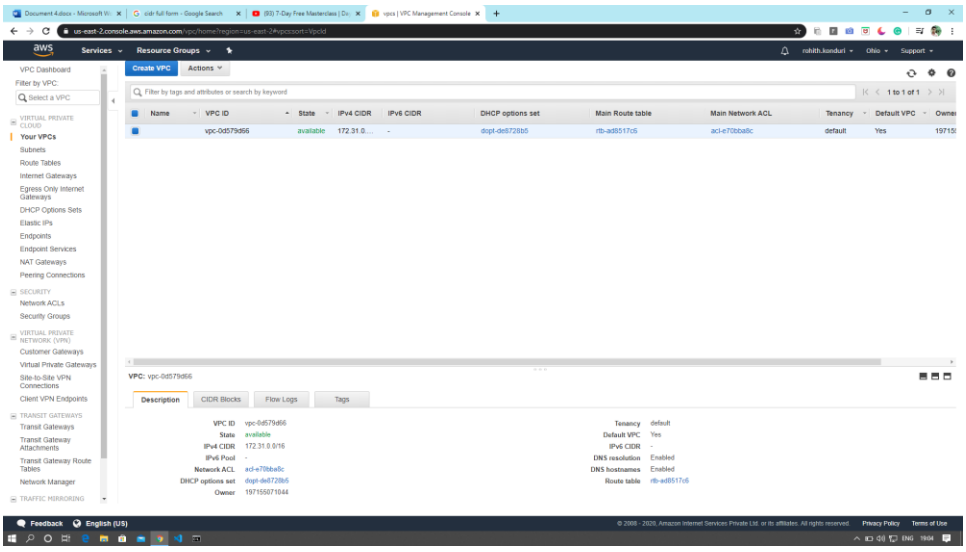
DAY 03

VPC.

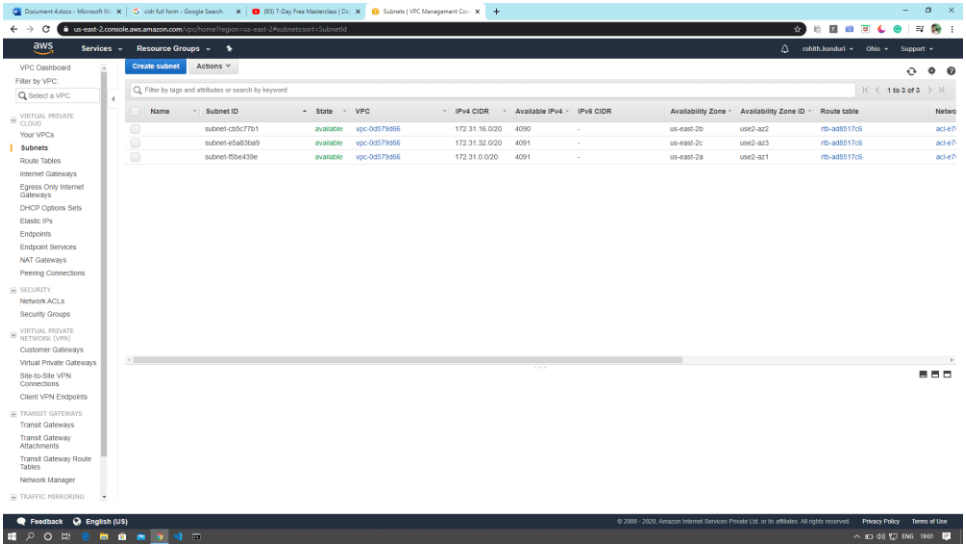
This is vpc dashboard.



Current VPCs

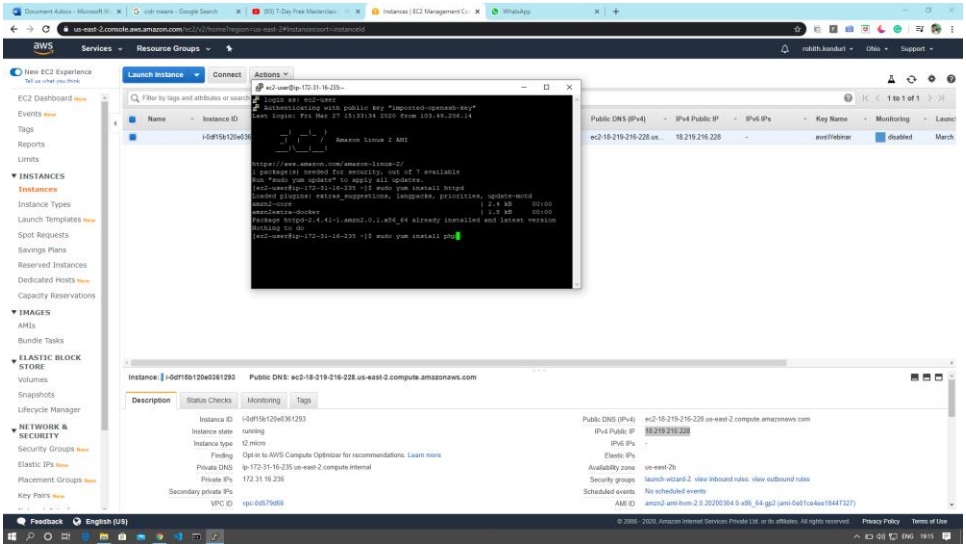


Number of subnets can be seen at subnet section.

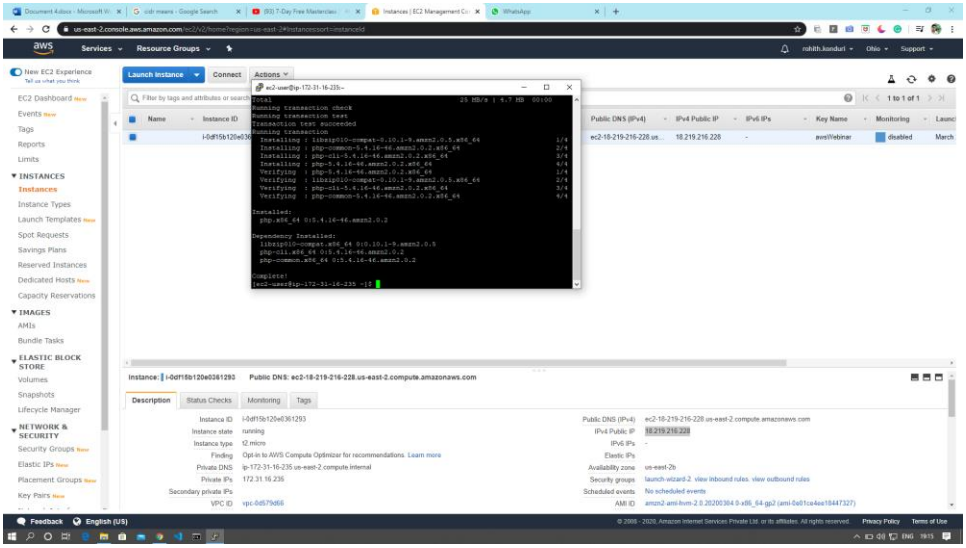


Using IP address of instance in putty we can login by user name “ec2-user”.

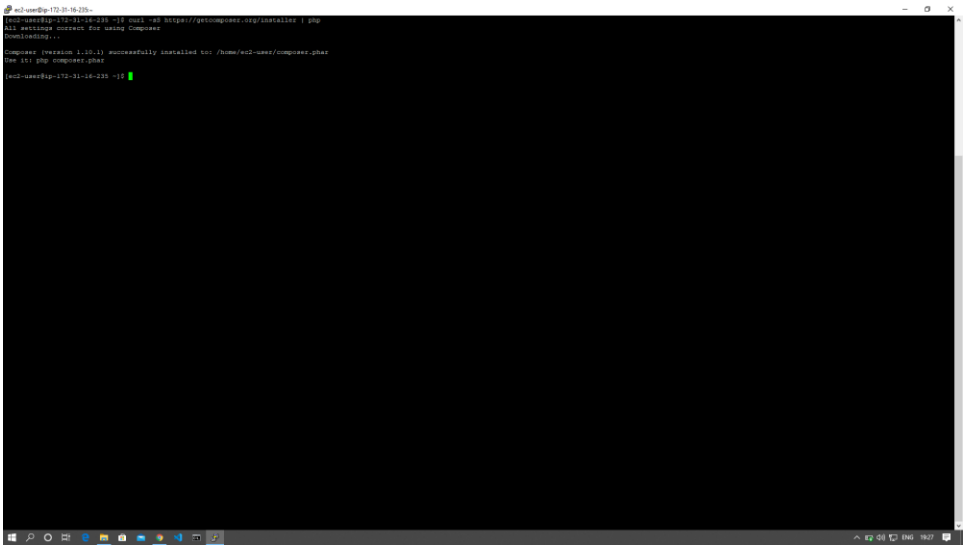
Now installing PHP.



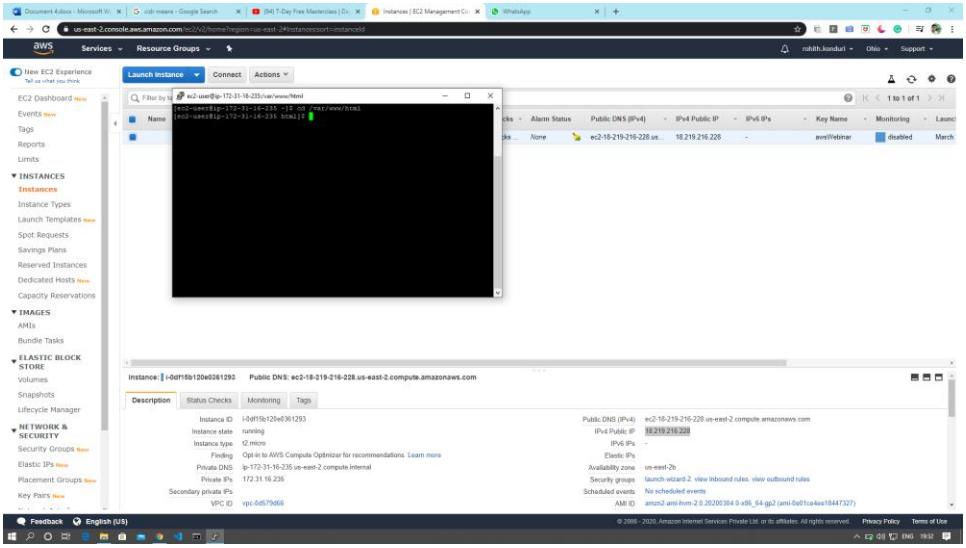
PHP installation completed.



Installing composer.

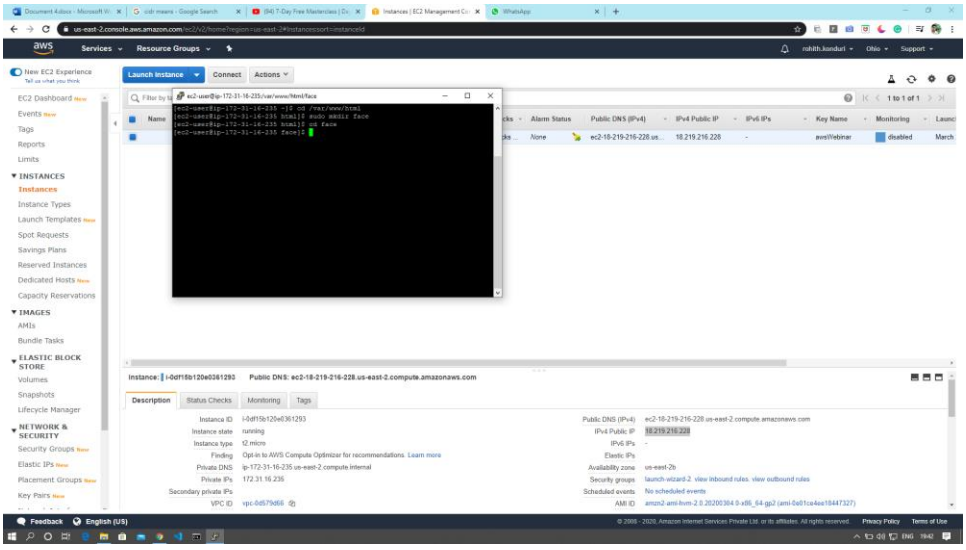


Changing directory.

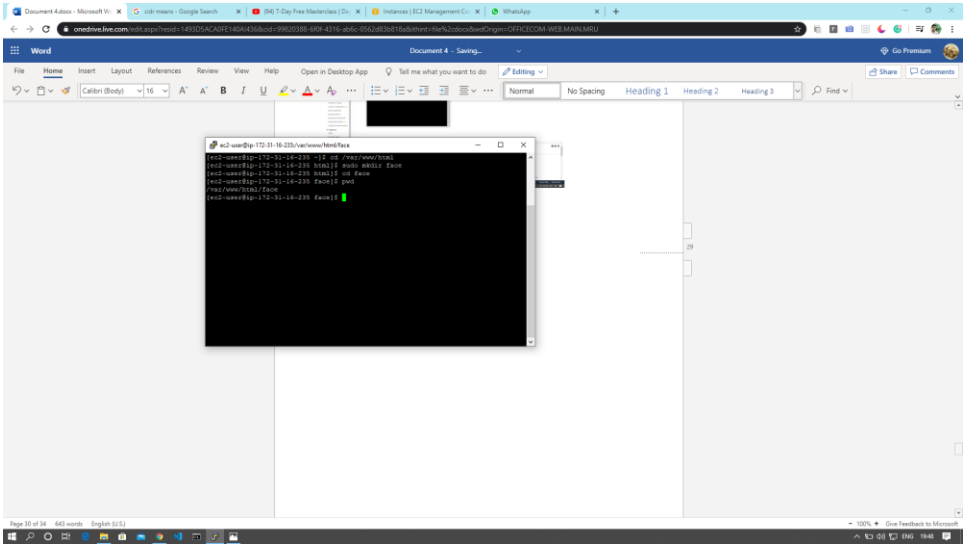


Making a new directory named “face” .

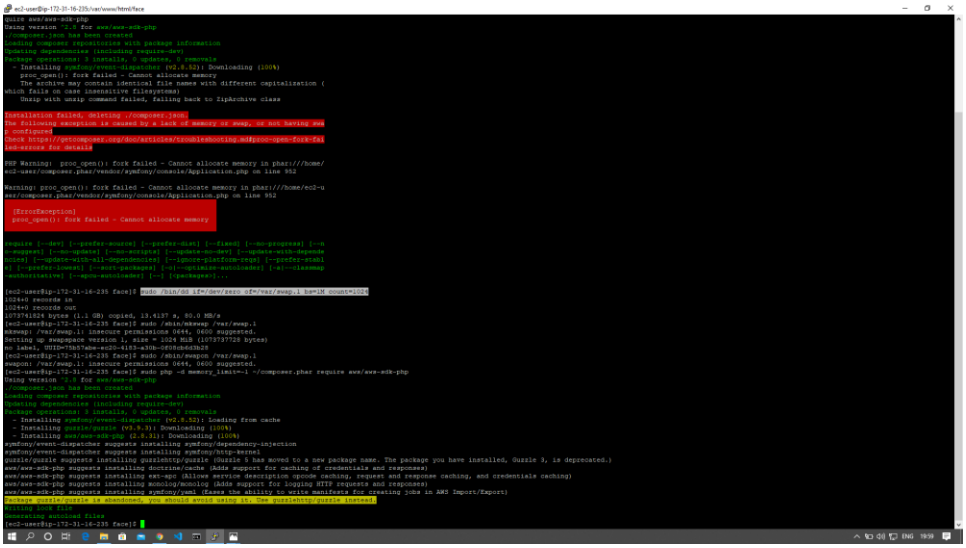
Changing directory to face:



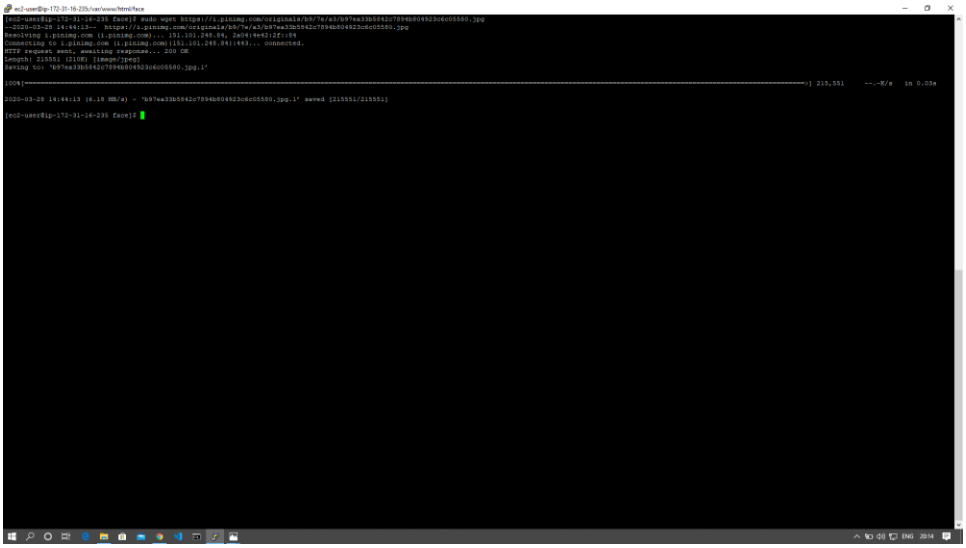
Present Working Directory to face :



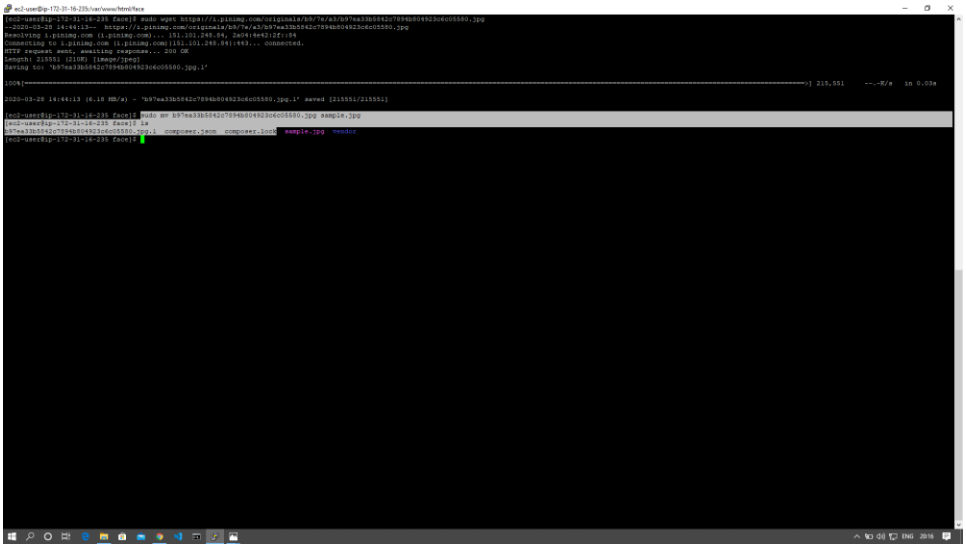
Installing Software Development Kit :



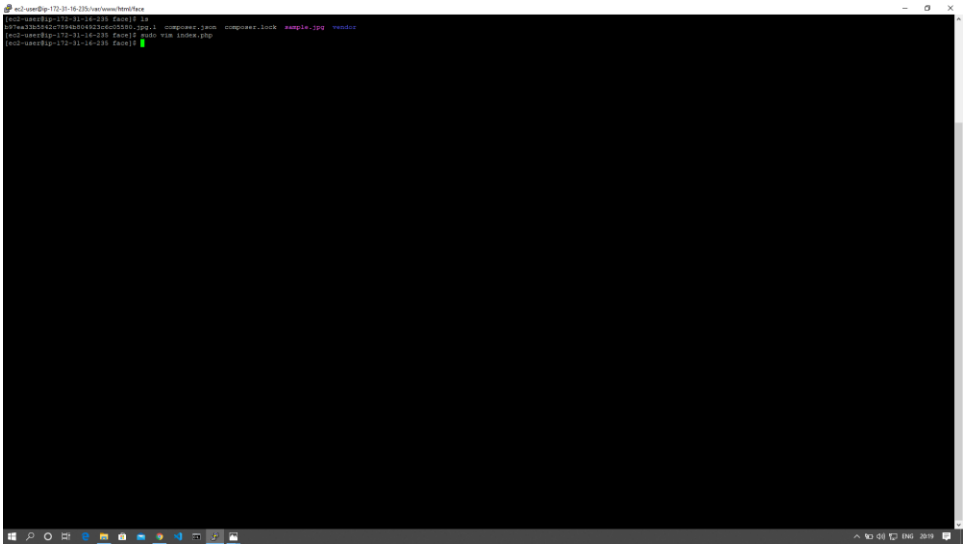
Saving image using image link :



Renaming jpg and downloading it :

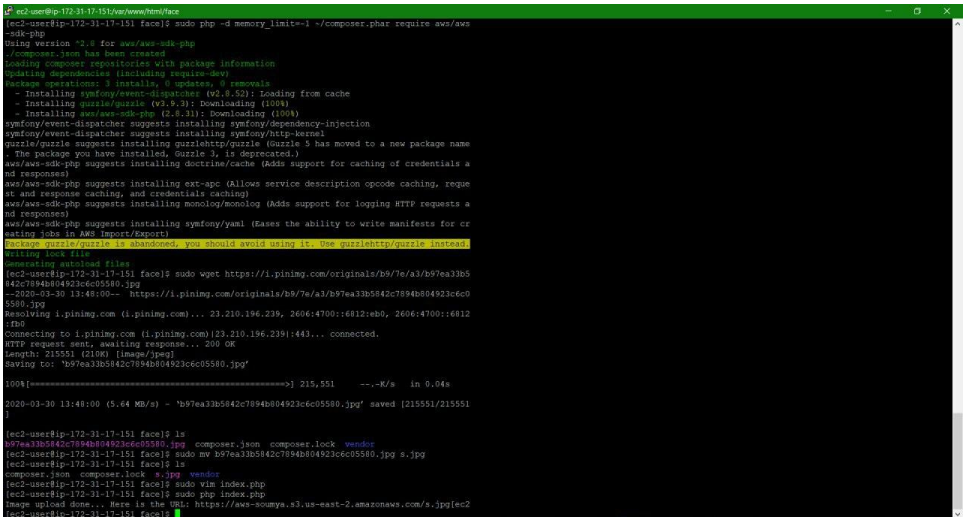


Copying the php code and pasting it in index.php in putty:



Pasting and esc, :wq :

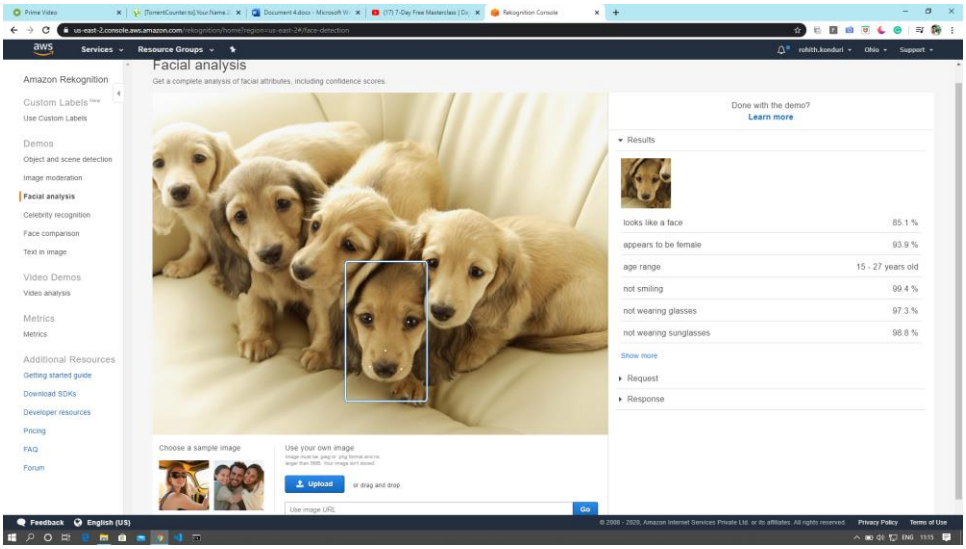
Uploading image to aws bucket:

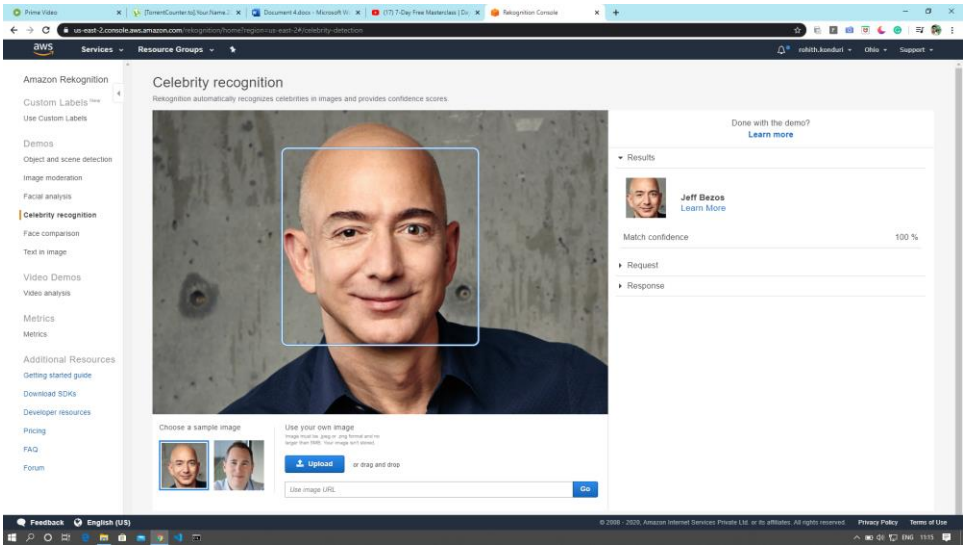


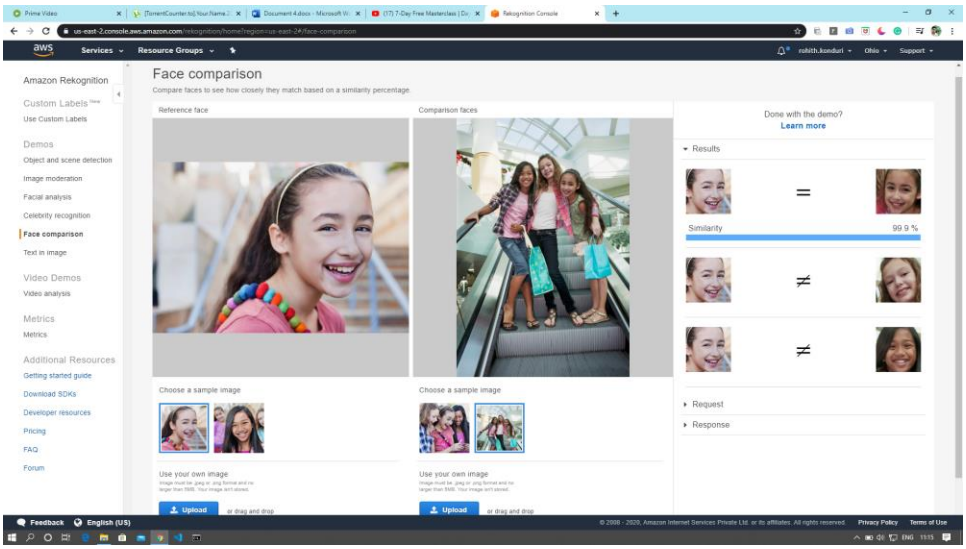
Face detection:

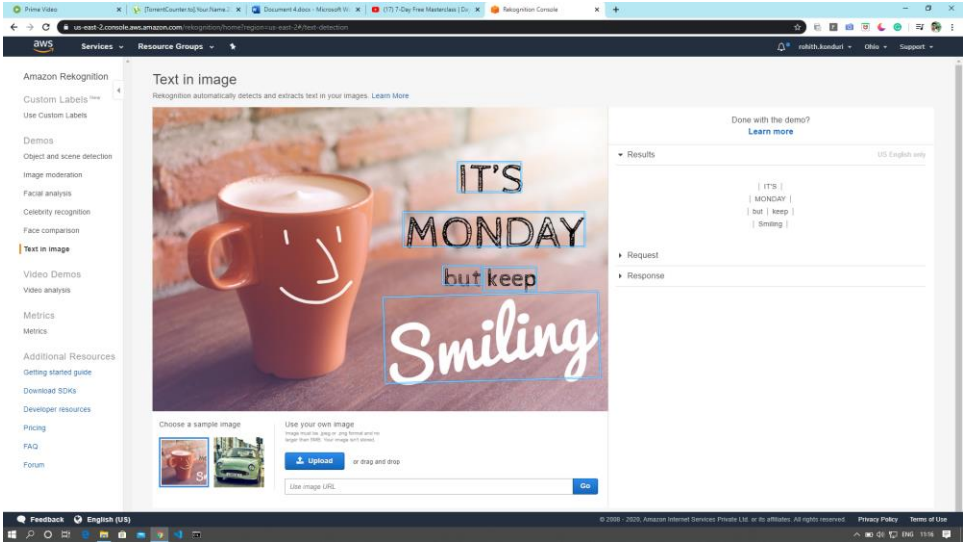
```
[ec2-user@ip-172-31-17-151 face]$ pwd
/var/www/html/face
[ec2-user@ip-172-31-17-151 face]$ sudo php -d memory_limit=-1 ~/composer.phar require aws/aws
-sdk-php
Using version ^3.133 for aws/aws-sdk-php
./composer.json has been updated
Loading composer repositories with package information
Updating dependencies (including require-dev)
Package operations: 7 installs, 1 update, 0 removals
  - Installing symfony/polyfill-mbstring (v1.15.0): Downloading (100%)
  - Installing mtdowling/jmespath.php (2.5.0): Downloading (100%)
  - Installing guzzlehttp/promises (v1.3.1): Downloading (100%)
  - Installing ralouphie/getallheaders (3.0.3): Downloading (100%)
  - Installing psr/http-message (1.0.1): Downloading (100%)
  - Installing guzzlehttp/psr7 (1.6.1): Downloading (100%)
  - Installing guzzlehttp/guzzle (6.5.2): Downloading (100%)
  - Updating aws/aws-sdk-php (2.8.31 => 3.133.46): Downloading (100%)
guzzlehttp/psr7 suggests installing zendframework/zend-httphandlerrunner (Emit PSR-7 response
s)
guzzlehttp/guzzle suggests installing psr/log (Required for using the Log middleware)
guzzlehttp/guzzle suggests installing ext-intl (Required for Internationalized Domain Name (I
DN) support)
Package guzzle/guzzle is abandoned, you should avoid using it. Use guzzlehttp/guzzle instead.
Writing lock file
Generating autoload files
1 package you are using is looking for funding.
Use the `composer fund` command to find out more!
[ec2-user@ip-172-31-17-151 face]$ sudo wget https://i.pinimg.com/originals/b9/7e/a3/b97ea33b5
842c7894b804923c6c05580.jpg
--2020-03-30 13:55:46--  https://i.pinimg.com/originals/b9/7e/a3/b97ea33b5842c7894b804923c6c0
5580.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 0.04s

2020-03-30 13:55:46 (4.66 MB/s) - 'b97ea33b5842c7894b804923c6c05580.jpg' saved [215551/215551
]

[ec2-user@ip-172-31-17-151 face]$ sudo mv b97ea33b5842c7894b804923c6c05580.jpg sample.jpg
[ec2-user@ip-172-31-17-151 face]$ ls
composer.json  composer.lock  index.php  sample.jpg  s.jpg  vendor
[ec2-user@ip-172-31-17-151 face]$ sudo vim index.php
[ec2-user@ip-172-31-17-151 face]$ sudo php index.php
Image upload done... Here is the URL: https://aws-soumya.s3.us-east-2.amazonaws.com/sample.jp
gTotally there are 9 faces[ec2-user@ip-172-31-17-151 face]$
```

Amazon rekognition :






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
99.9 %
Request
Response
Choose a sample image
Use your own image
Upload
or drag and drop


Text in image
Rekognition automatically detects and extracts text in your images. Learn More
IT'S
MONDAY
but keep
Smiling
Done with the demo?
Learn more
Results
Request
Response
Choose a sample image
Use your own image
Upload
or drag and drop
Use image URL
Go