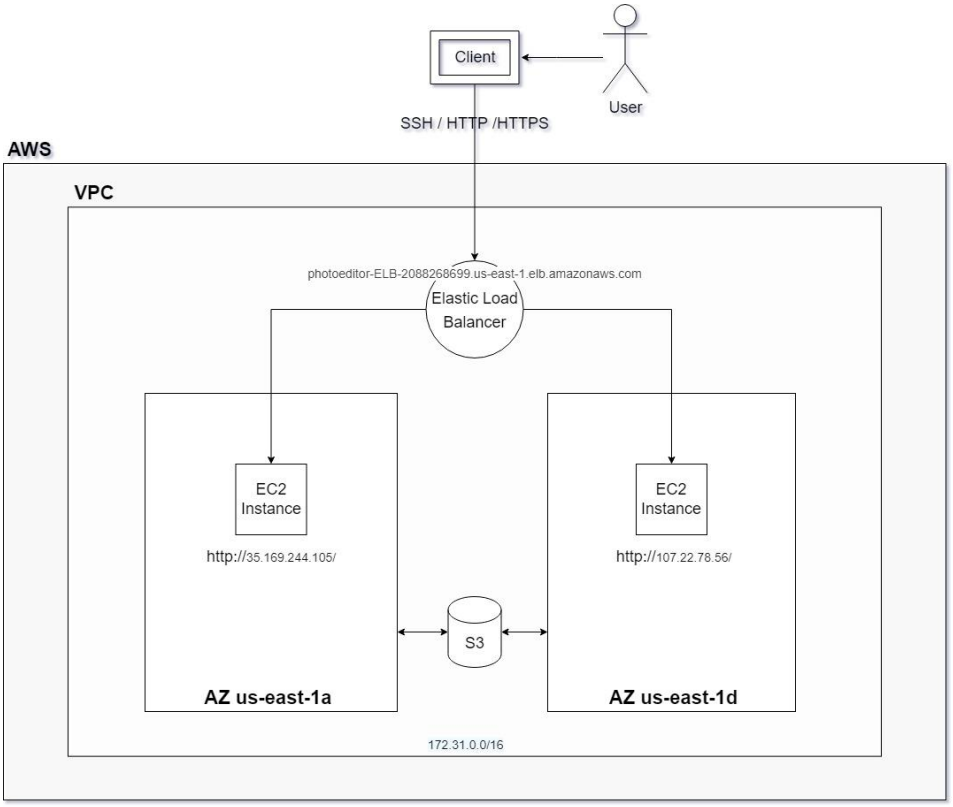
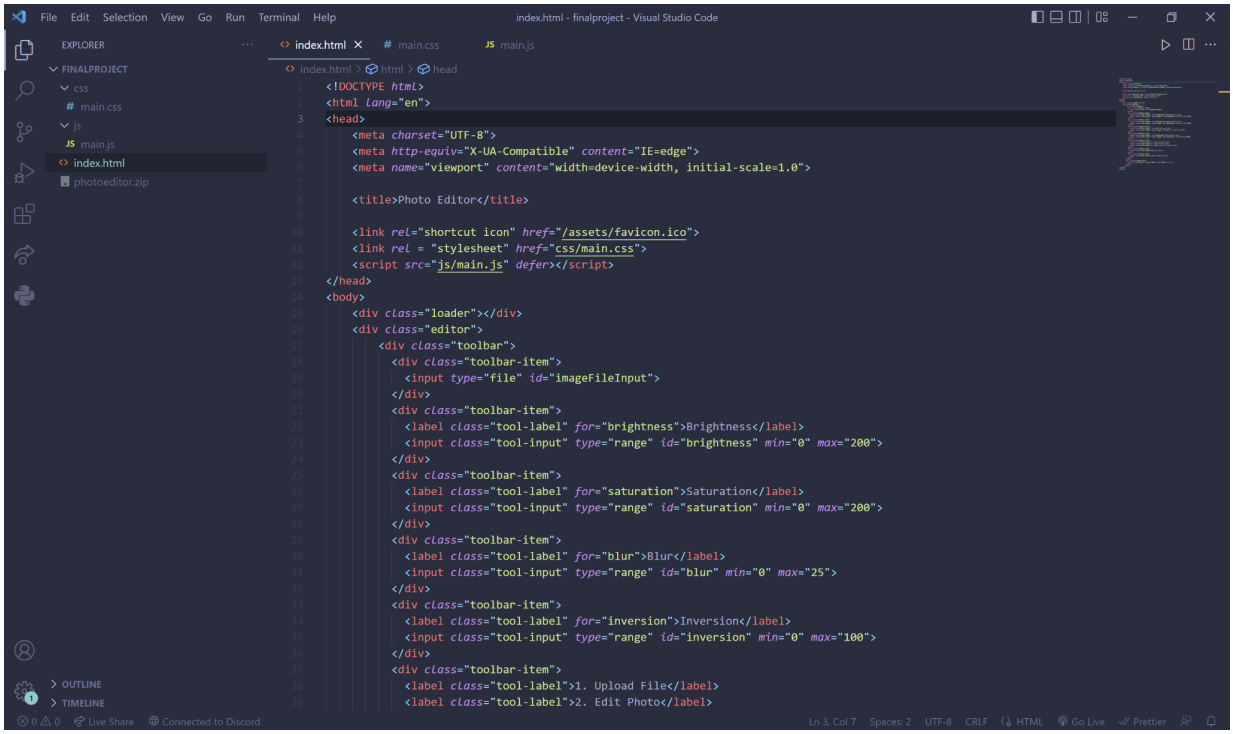
Gouw Jordan Jonathan
32209137
Cloud Final Report

Since my hobby is to take pictures and edit them, I got the idea to be making a simple photo image editor for this cloud final project. And so I've made a diagram to show what my aws infrastructure looks like, so that it could be easier understood.



First, I made the website or web layout for the photo editor on vsc and I will use html, css, and javascript. I also use Xampp to test locally on my browser to see if my photo editor works just fine before importing it to S3 on AWS, because it will be a bit difficult to make changes afterwards.

ELB : photoeditor-ELB-2088268699.us-east-1.elb.amazonaws.com
Elastic IP 1 : http://107.22.78.56/
Elastic IP 2 : http://35.169.244.105/


File Edit Selection View Go Run Terminal Help
index.html - finalproject - Visual Studio Code
EXPLORER
FINALPROJECT
css
main.css
js
main.js
index.html
photoeditor.zip
index.html
main.css
main.js
index.html > html > head
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Photo Editor</title>
<link rel="shortcut icon" href="/assets/favicon.ico">
<link rel = "stylesheet" href="css/main.css">
<script src="js/main.js" defer></script>
</head>
<body>

<input type="file" id="imageFileInput">

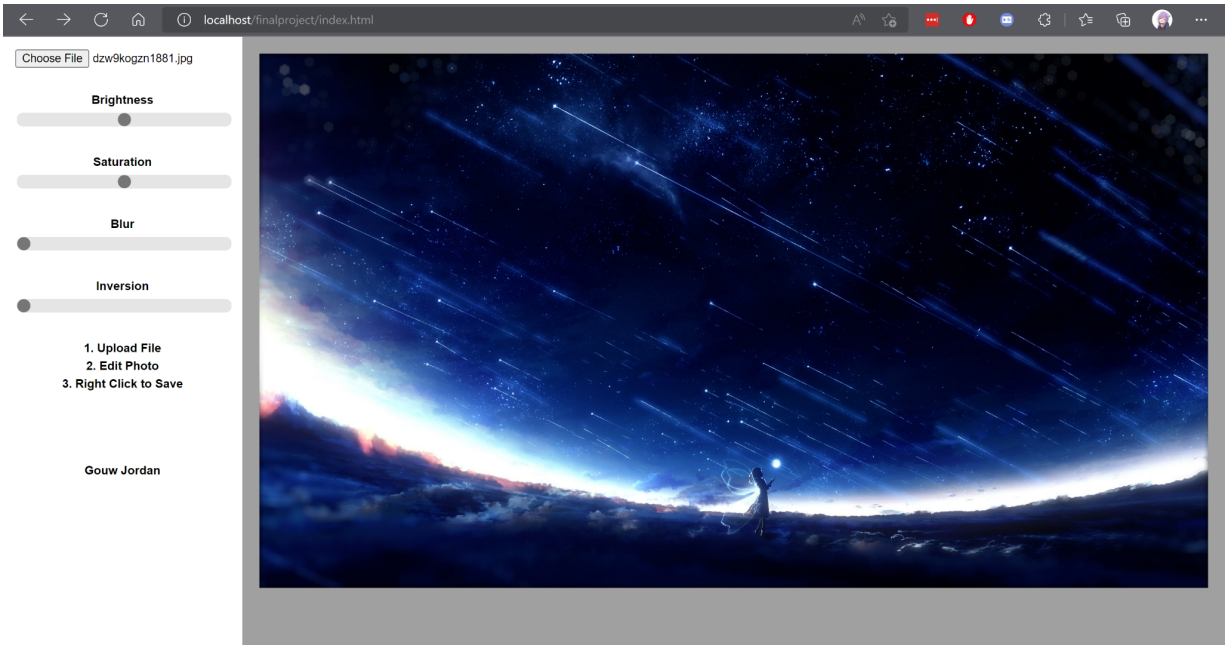
<label class="tool-label" for="brightness">Brightness</label>
<input class="tool-input" type="range" id="brightness" min="0" max="200">

<label class="tool-label" for="saturation">Saturation</label>
<input class="tool-input" type="range" id="saturation" min="0" max="200">

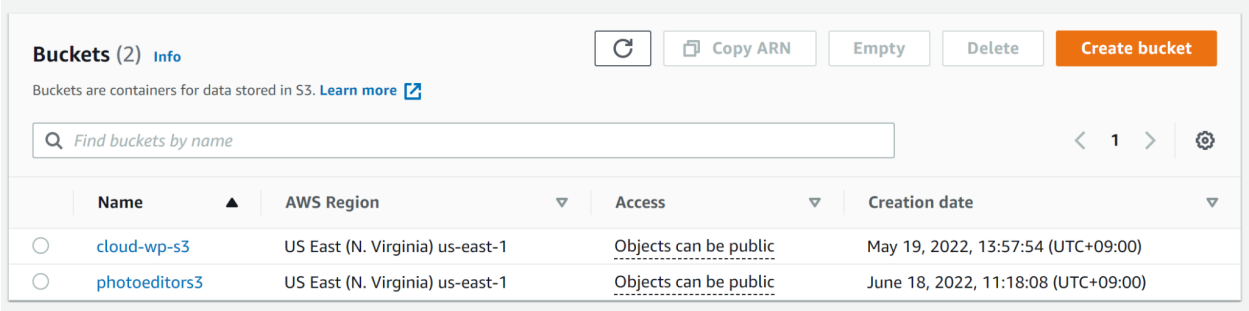
<label class="tool-label" for="blur">Blur</label>
<input class="tool-input" type="range" id="blur" min="0" max="25">

<label class="tool-label" for="inversion">Inversion</label>
<input class="tool-input" type="range" id="inversion" min="0" max="100">

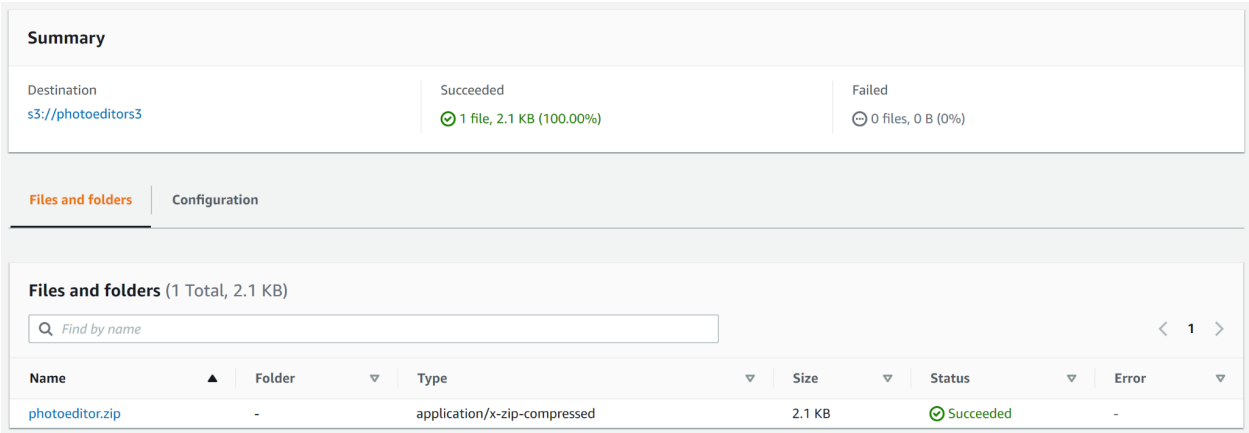
<label class="tool-label">1. Upload File</label>
<label class="tool-label">2. Edit Photo</label>
OUTLINE
TIMELINE
Live Share
Connected to Discord
Ln 3, Col 7
Spaces: 2
UTF-8
CRLF
HTML
Go Live
Prettier


localhost/finalproject/index.html
Choose File
dzw9kogzn1881.jpg
Brightness
Saturation
Blur
Inversion
1. Upload File
2. Edit Photo
3. Right Click to Save
Gouw Jordan

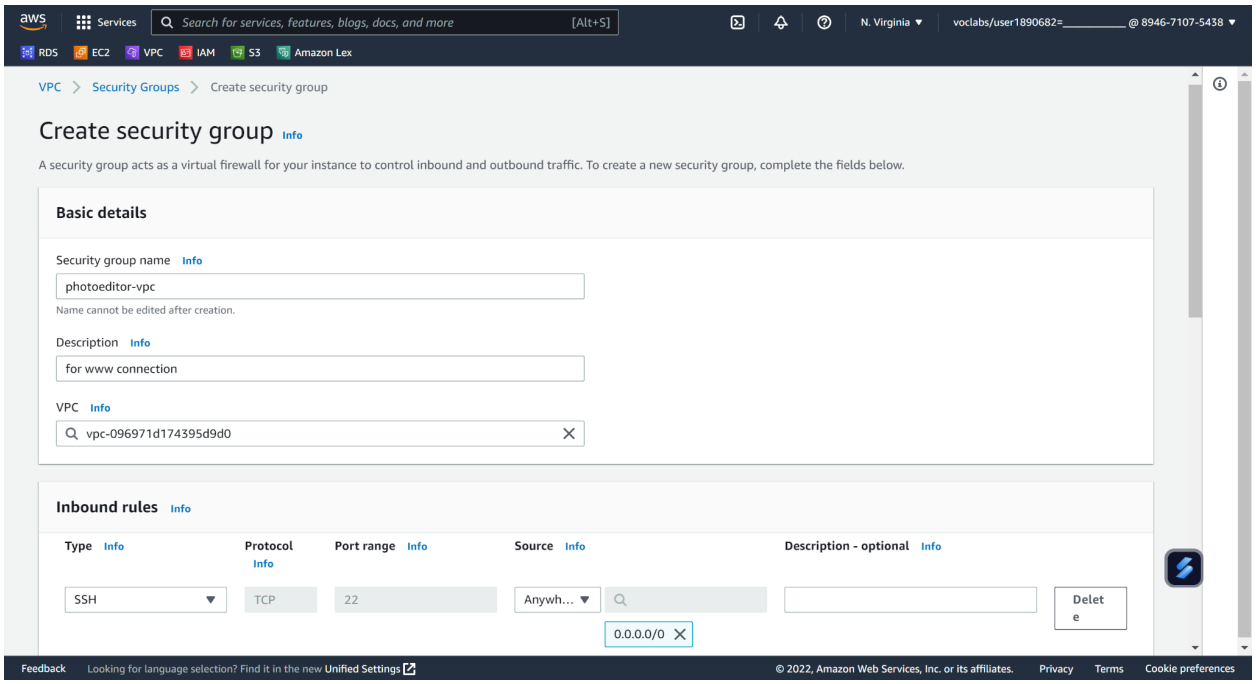
I made an S3 bucket and named it photoeditors3 and set it to public so that later i can extract my web code files from s3 to the instance

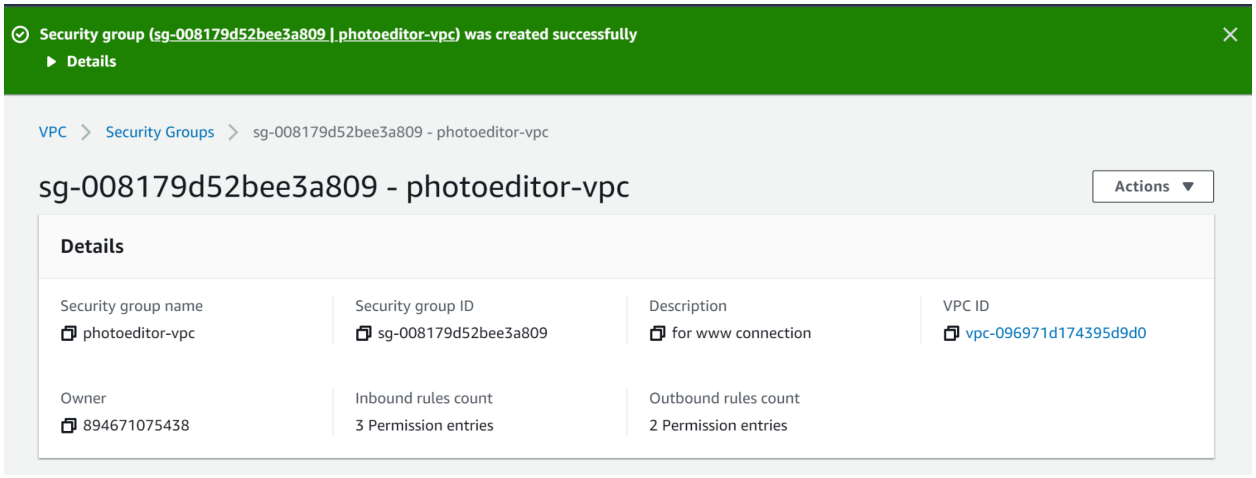


And inside the photoeditors3 S3 i will upload my website that has been zipped to be extracted later.

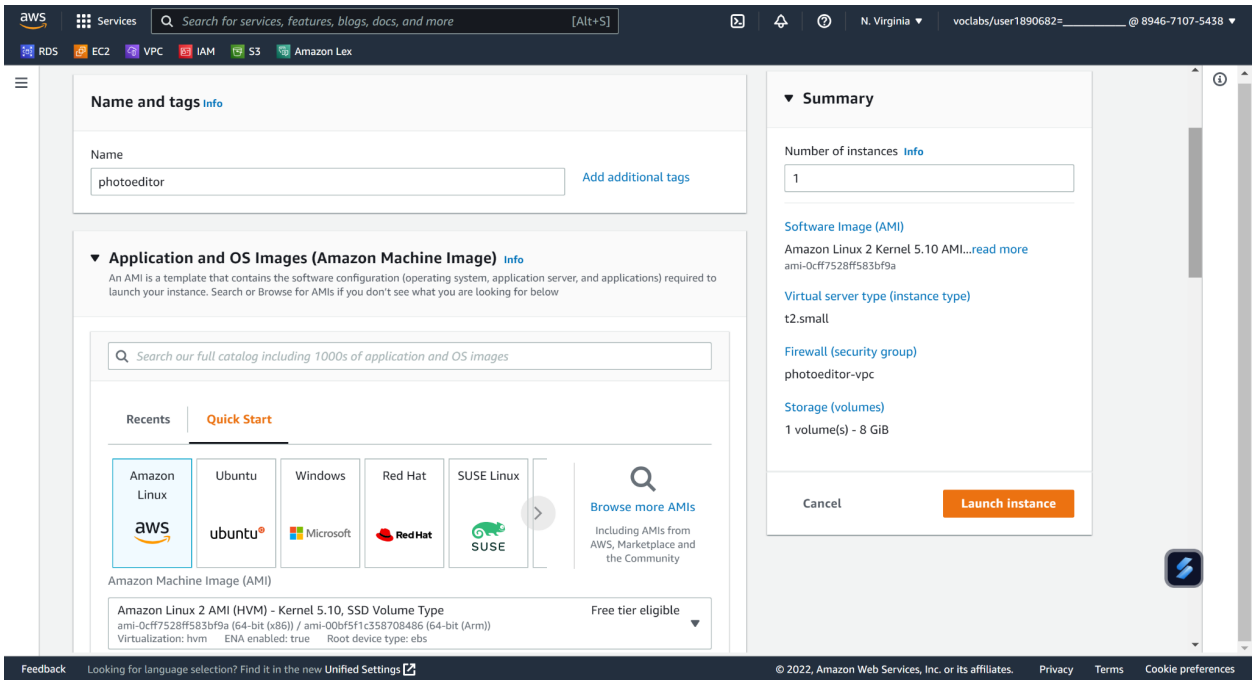


I will be using a new VPC and not the default one. For the VPC i will name it photoeditor-vpc and set the inbound rules to ssh, http, and https and set the source to anywhere or 0.0.0.0/0





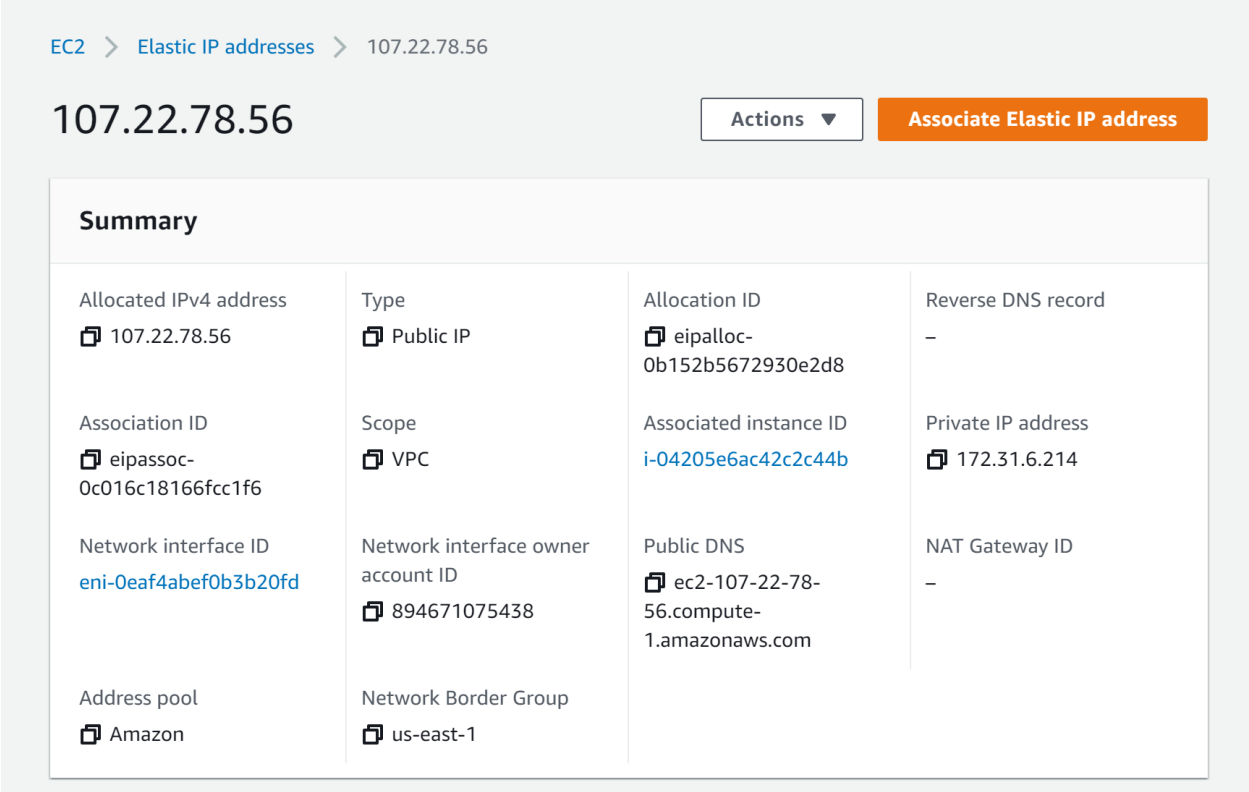
For the instance, I will be using amazon linux 2 AMI and change the VPC using the one that I have created previously named photoeditor-vpc



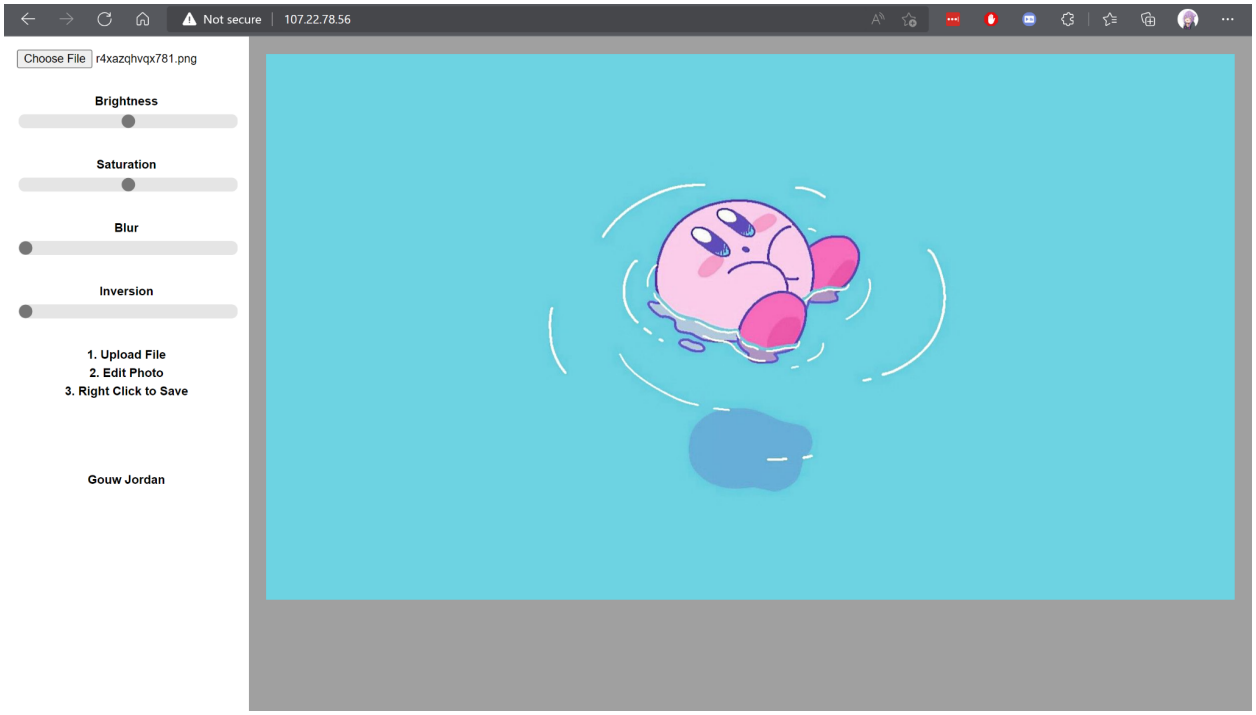
Then inside the photo editor instance I will connect using ssh provided by aws and not xshell. Inside I will try to extract to unzip the files from s3 bucket using its object url (https://photoeditors3.s3.amazonaws.com/photoeditor.zip) .

```
[root@ip-172-31-6-214 ec2-user]# pwd
/home/ec2-user
[root@ip-172-31-6-214 ec2-user]# cd /var/www/html
[root@ip-172-31-6-214 html]# wget https://photoeditors3.s3.amazonaws.com/photoeditor.zip
--2022-06-18 02:48:32--  https://photoeditors3.s3.amazonaws.com/photoeditor.zip
Resolving photoeditors3.s3.amazonaws.com (photoeditors3.s3.amazonaws.com)... 54.231.200.105
Connecting to photoeditors3.s3.amazonaws.com (photoeditors3.s3.amazonaws.com)|54.231.200.105|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2407 (2.4K) [application/zip]
Saving to: 'photoeditor.zip'

100%[==========================================================================================================>] 2,407       --.-K/s   in 0s

2022-06-18 02:48:33 (32.7 MB/s) - 'photoeditor.zip' saved [2407/2407]

[root@ip-172-31-6-214 html]# ls
photoeditor.zip
[root@ip-172-31-6-214 html]# unzip photoeditor.zip
Archive:  photoeditor.zip
   creating: photoeditor/css/
  inflating: photoeditor/css/main.css
  inflating: photoeditor/index.html
   creating: photoeditor/js/
  inflating: photoeditor/js/main.js
[root@ip-172-31-6-214 html]# ls
photoeditor  photoeditor.zip
[root@ip-172-31-6-214 html]# mv photoeditor/* .
[root@ip-172-31-6-214 html]# ls
css  index.html  js  photoeditor  photoeditor.zip
[root@ip-172-31-6-214 html]# service httpd start
Redirecting to /bin/systemctl start httpd.service
[root@ip-172-31-6-214 html]#
```

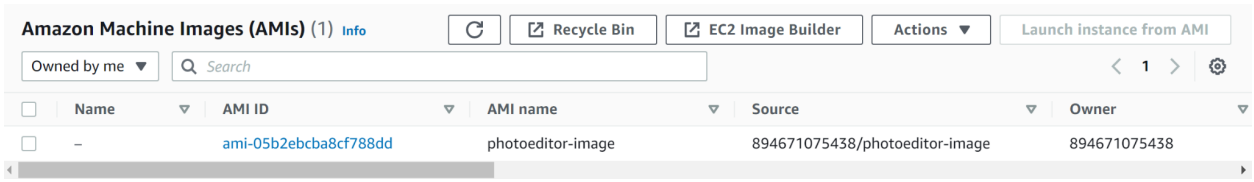
I will be making 2 elastic IP for the 2 instances that i will be making so that we don't need to manually change the ip that changes every time we start the lab.



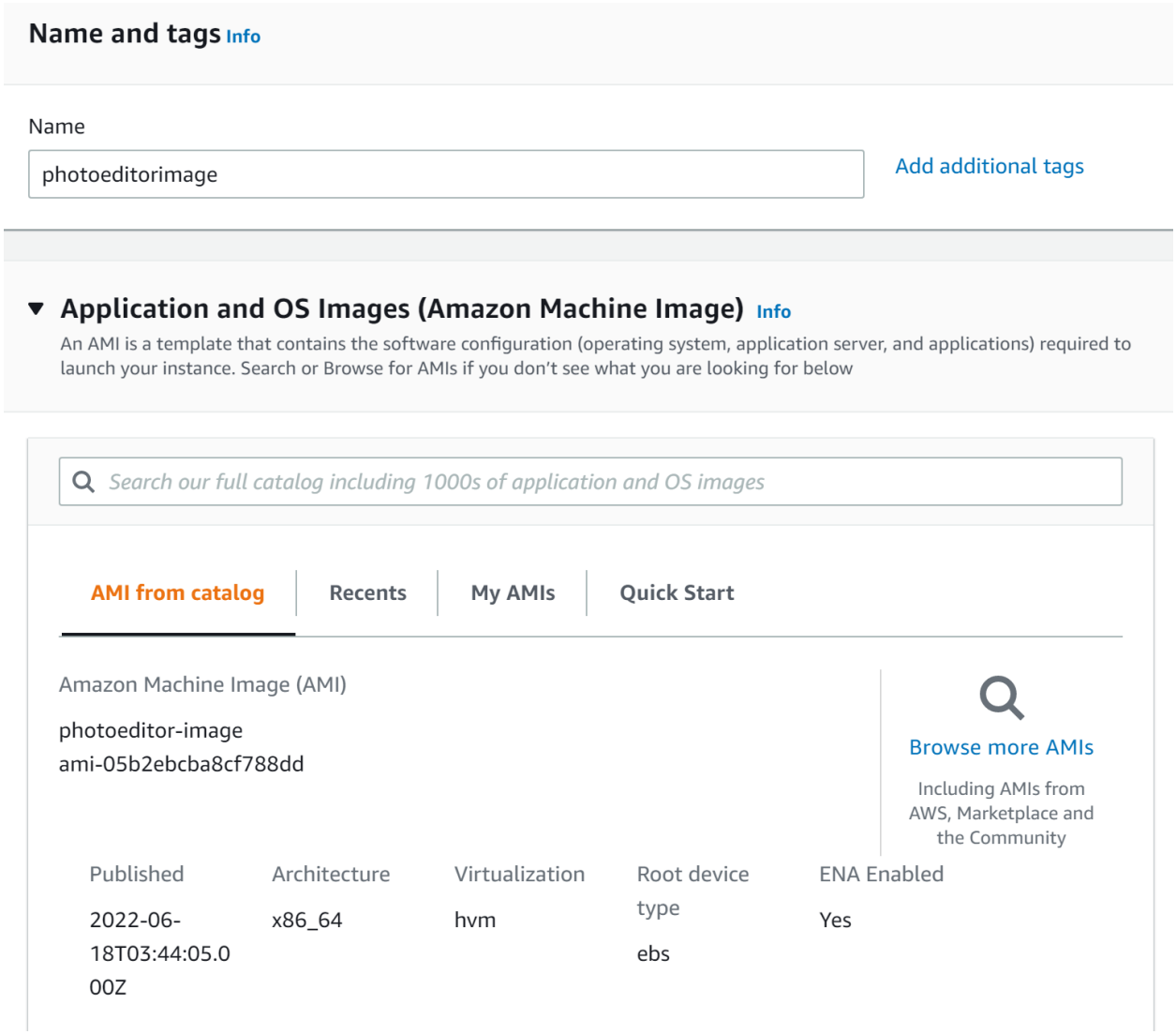
For the 1st instance i will allocate the elastic IP of http://107.22.78.56/ and it works



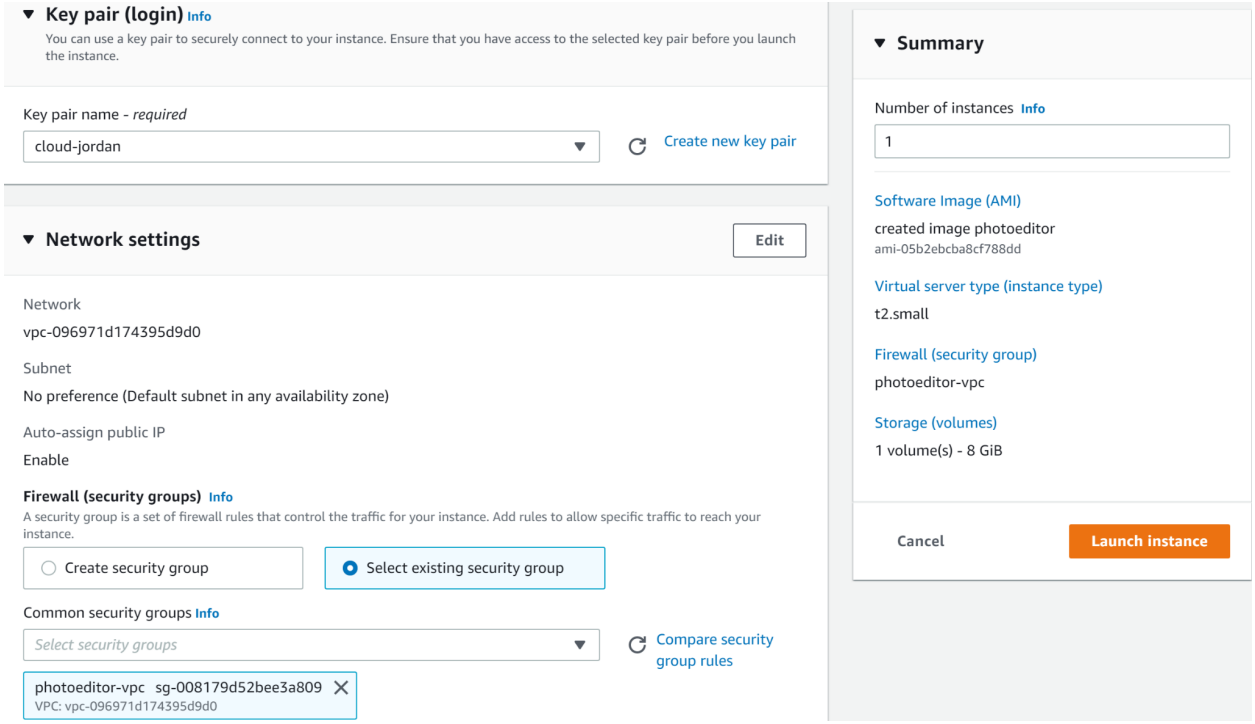
Then from the first instance that i have created i will create an image so that we don't have to set up from zero again.



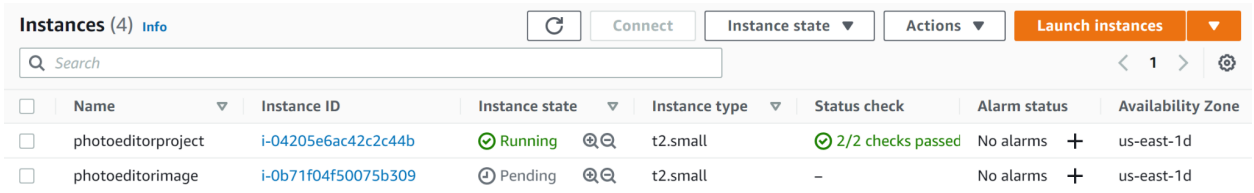
Then I will try to launch an instance named photoeditor_image for the second instance created using the image I have made from the first instance photoeditorproject.



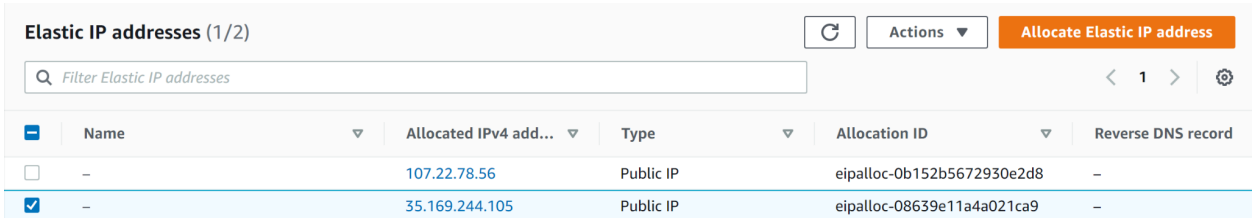
The setting for the second instance created from the image. I will be using the existing security group and vpc.



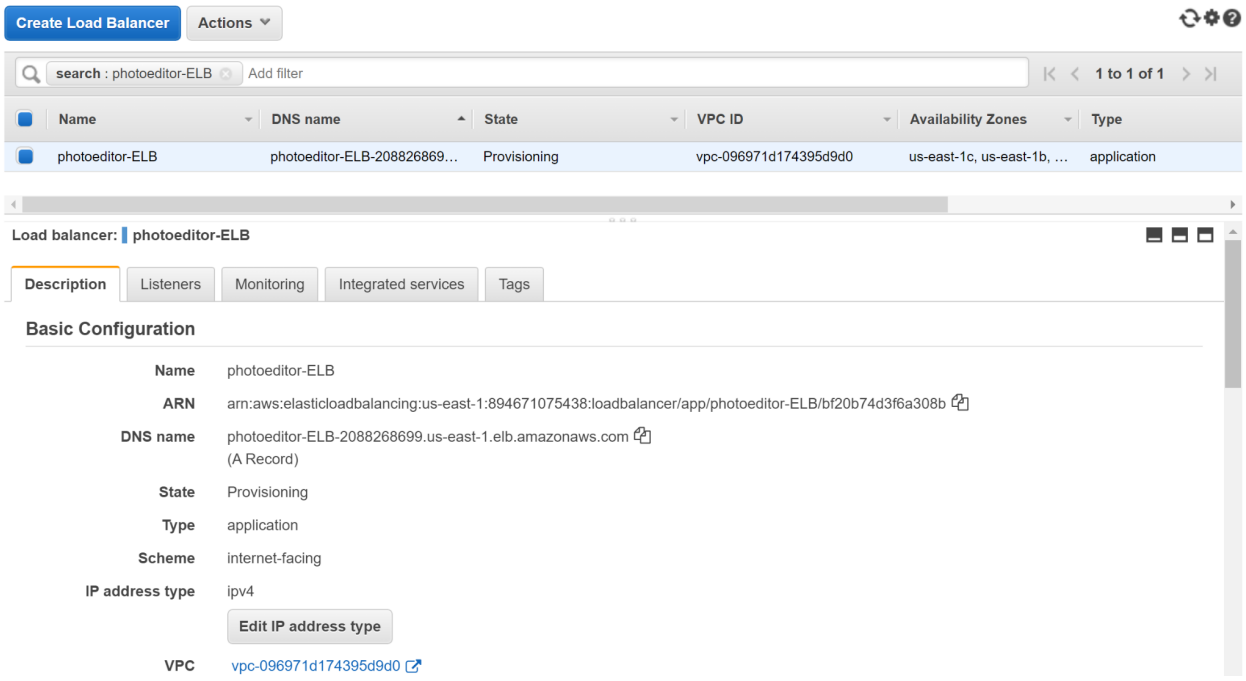
These will be 2 instances used in this project named photoeditorproject and photoeditorimage.



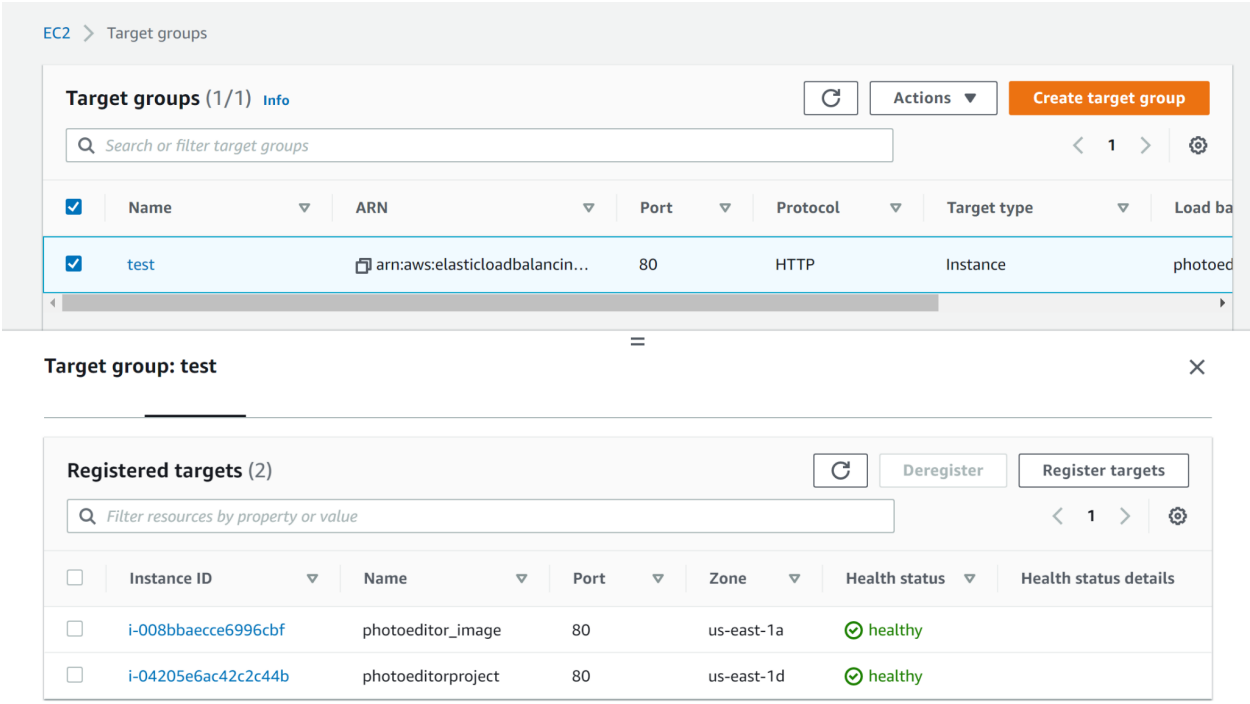
I will assign the elastic IP of http://35.169.244.105/ to the new instance created from the image.



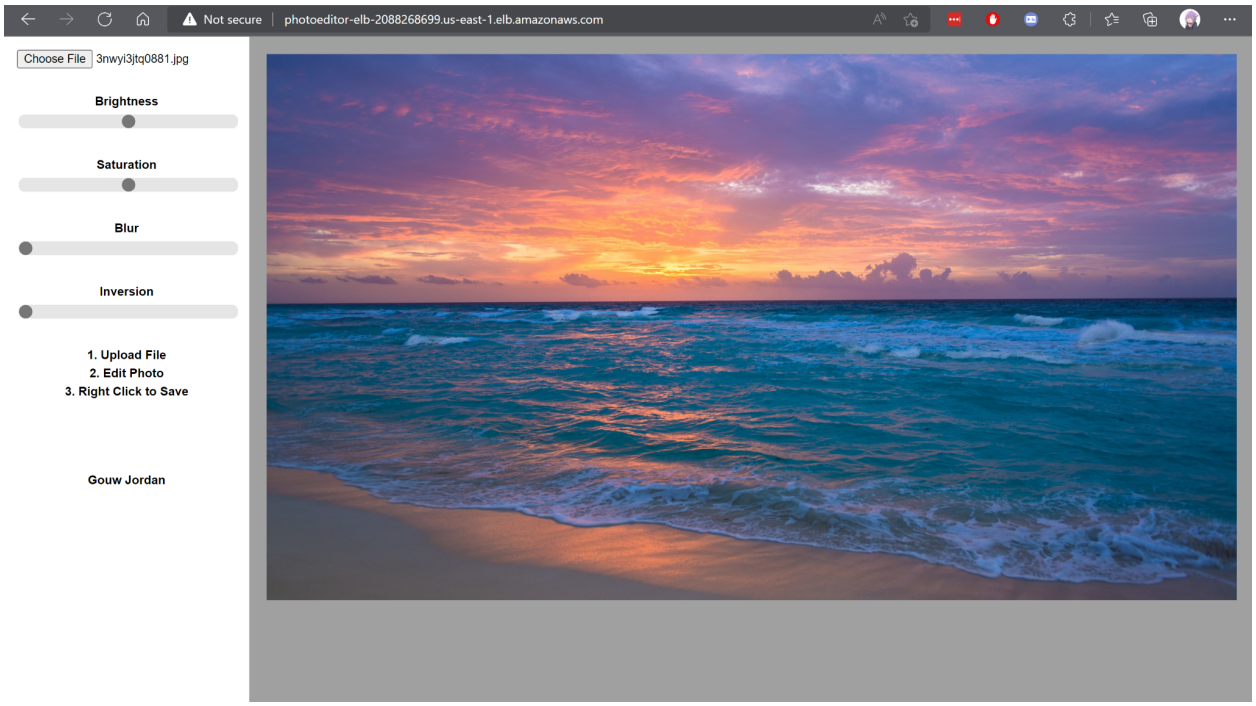
I will be making ELB that will do dynamic distribution from my two instances.



And I have created a target group and registered my two instances to the target group for the elb to distribute.



And using the ELB dns name (photoeditor-ELB-2088268699.us-east-1.elb.amazonaws.com)
The ELB or Load Balancer Works!



Lastly I also did a little benchmark on one of my instances to see how it fares.

```
Benchmarking 107.22.78.56 (be patient)
Completed 5000 requests
Completed 10000 requests
Completed 15000 requests
Completed 20000 requests
Finished 20434 requests


Server Software:        Apache/2.4.53
Server Hostname:        107.22.78.56
Server Port:            80

Document Path:          /
Document Length:        2100 bytes

Concurrency Level:      500
Time taken for tests:   10.000 seconds
Complete requests:      20434
Failed requests:        0
Total transferred:      48845776 bytes
HTML transferred:       43026900 bytes
Requests per second:    2043.40 [#/sec] (mean)
Time per request:       244.691 [ms] (mean)
Time per request:       0.489 [ms] (mean, across all concurrent requests)
Transfer rate:          4770.09 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0   19  11.8     18      65
Processing:    10  223  81.5    182     414
Waiting:        1  215  86.1    169     414
Total:         10  242  73.3    210     426

Percentage of the requests served within a certain time (ms)
  50%    210
  66%    243
  75%    310
  80%    339
  90%    360
  95%    372
  98%    389
  99%    396
 100%    426 (longest request)
```

Thank you…!