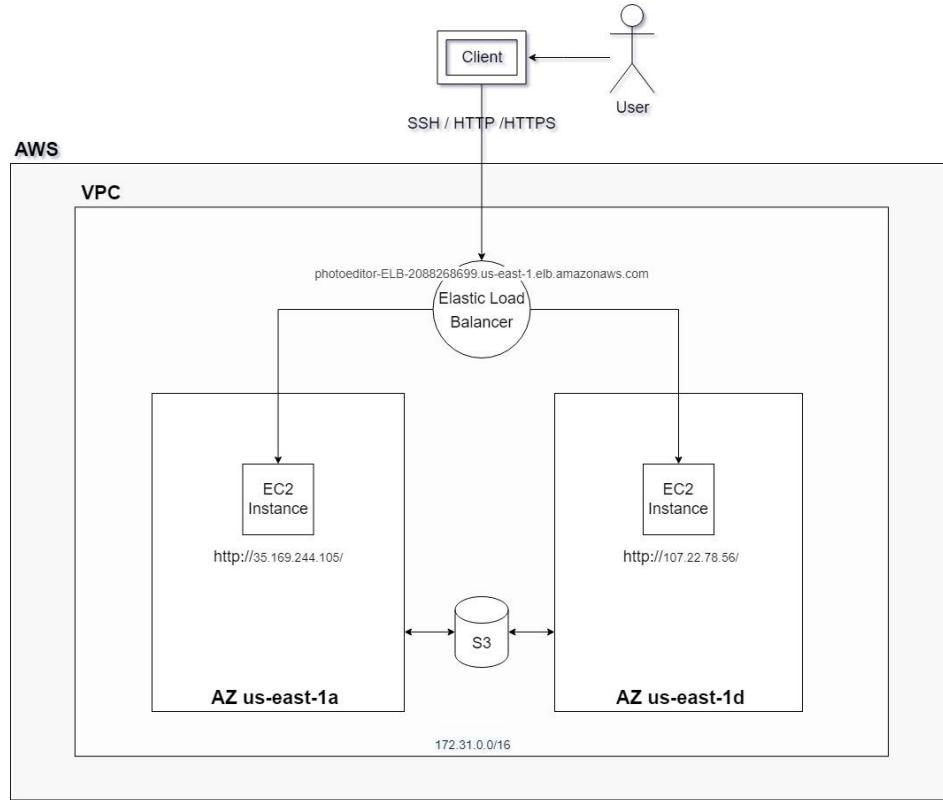


Gouw Jordan Jonathan
32209137
Cloud Final Report

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

index.html # main.css main.js

```
<!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>
```

index.html photoeditor.zip

index.html > html > head

14 <head>

15 <div class="loader"></div>

16 <div class="editor">

17 <div class="toolbar">

18 <div class="toolbar-item">

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

20 </div>

21 <div class="toolbar-item">

22 <label class="tool-label" for="brightness">Brightness</label>

23 <input class="tool-input" type="range" id="brightness" min="0" max="200">

24 </div>

25 <div class="toolbar-item">

26 <label class="tool-label" for="saturation">Saturation</label>

27 <input class="tool-input" type="range" id="saturation" min="0" max="200">

28 </div>

29 <div class="toolbar-item">

30 <label class="tool-label" for="blur">Blur</label>

31 <input class="tool-input" type="range" id="blur" min="0" max="25">

32 </div>

33 <div class="toolbar-item">

34 <label class="tool-label" for="inversion">Inversion</label>

35 <input class="tool-input" type="range" id="inversion" min="0" max="100">

36 </div>

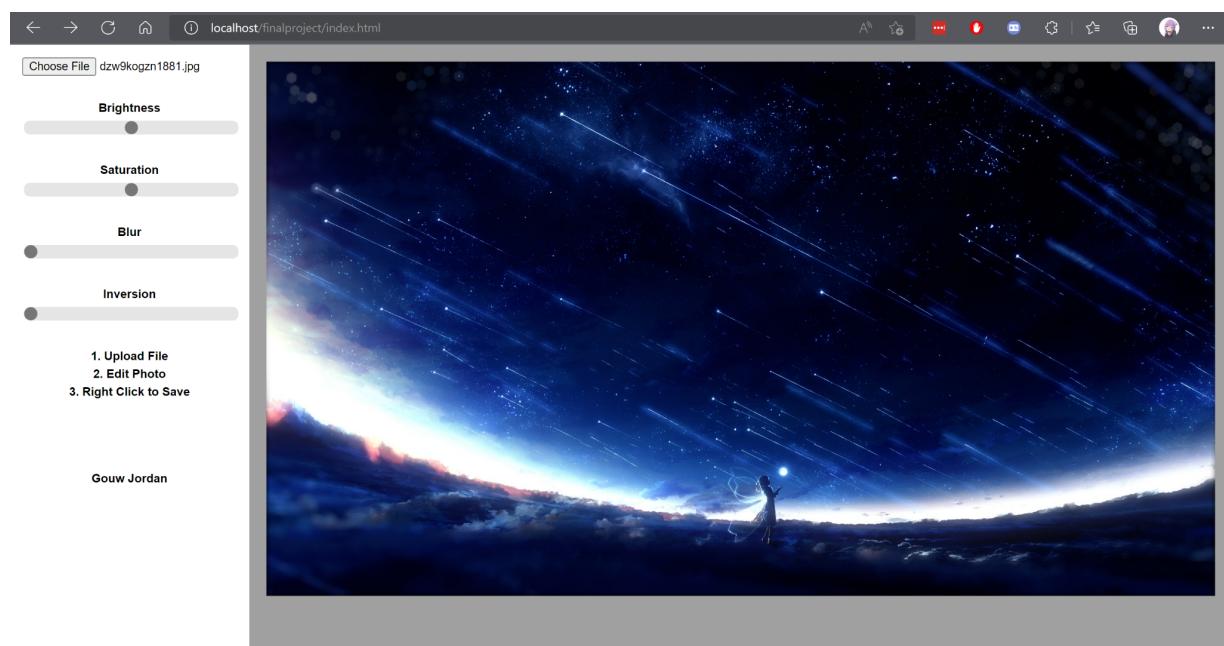
37 <div class="toolbar-item">

38 <label class="tool-label">1. Upload File</label>

39 <label class="tool-label">2. Edit Photo</label>

Live Share Connected to Discord

In 3 Col 7 Spacing 2 UTF-8 CR LF ⌘ HTML ⌘ Go Live ⌘ Prettier



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

The screenshot shows the AWS S3 Buckets list interface. At the top, there are buttons for 'Create bucket' (orange), 'Empty', 'Delete', 'Copy ARN', and a 'C' icon. Below this is a search bar with placeholder text 'Find buckets by name'. A table lists the buckets:

Name	AWS Region	Access	Creation date
cloud-wp-s3	US East (N. Virginia) us-east-1	Objects can be public	May 19, 2022, 13:57:54 (UTC+09:00)
photoeditors3	US East (N. Virginia) us-east-1	Objects can be public	June 18, 2022, 11:18:08 (UTC+09:00)

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

The screenshot shows the AWS Lambda deployment summary. It includes a 'Summary' section with deployment details and tabs for 'Files and folders' and 'Configuration'. Under 'Files and folders', it shows a single file uploaded:

Files and folders (1 Total, 2.1 KB)				
Name	Type	Size	Status	Error
photoeditor.zip	application/x-zip-compressed	2.1 KB	Succeeded	-

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

The screenshot shows the AWS VPC Security Groups creation interface. In the 'Basic details' section, the security group name is 'photoeditor-vpc' and the description is 'for www connection'. A VPC named 'vpc-096971d174395d9d0' is selected. In the 'Inbound rules' section, there is one rule: an SSH rule on port 22 from anywhere (0.0.0.0/0) to the security group itself. A success message at the bottom states: 'Security group (sg-008179d52bee3a809 | photoeditor-vpc) was created successfully'. The final view shows the details of the created security group 'sg-008179d52bee3a809 - photoeditor-vpc'.

Basic details

Security group name [Info](#)
photoeditor-vpc
Name cannot be edited after creation.

Description [Info](#)
for www connection

VPC [Info](#)
vpc-096971d174395d9d0

Inbound rules [Info](#)

Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info
SSH	TCP	22	Anywhere ▼	0.0.0.0/0 X

Feedback Looking for language selection? Find it in the new Unified Settings [?](#)

© 2022, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

sg-008179d52bee3a809 - photoeditor-vpc

Details

Security group name photoeditor-vpc	Security group ID sg-008179d52bee3a809	Description for www connection	VPC ID vpc-096971d174395d9d0
Owner 894671075438	Inbound rules count 3 Permission entries	Outbound rules count 2 Permission entries	

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

The screenshot shows the AWS CloudFormation console with a stack named 'photoeditor' in progress. The 'Template' tab is selected, displaying the CloudFormation template. The 'Outputs' tab shows the output 'photoeditor' with the value 'arn:aws:lambda:us-east-1:123456789012:function:photoeditor'. Other tabs like 'Logs', 'Events', and 'Resources' are also visible.

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.

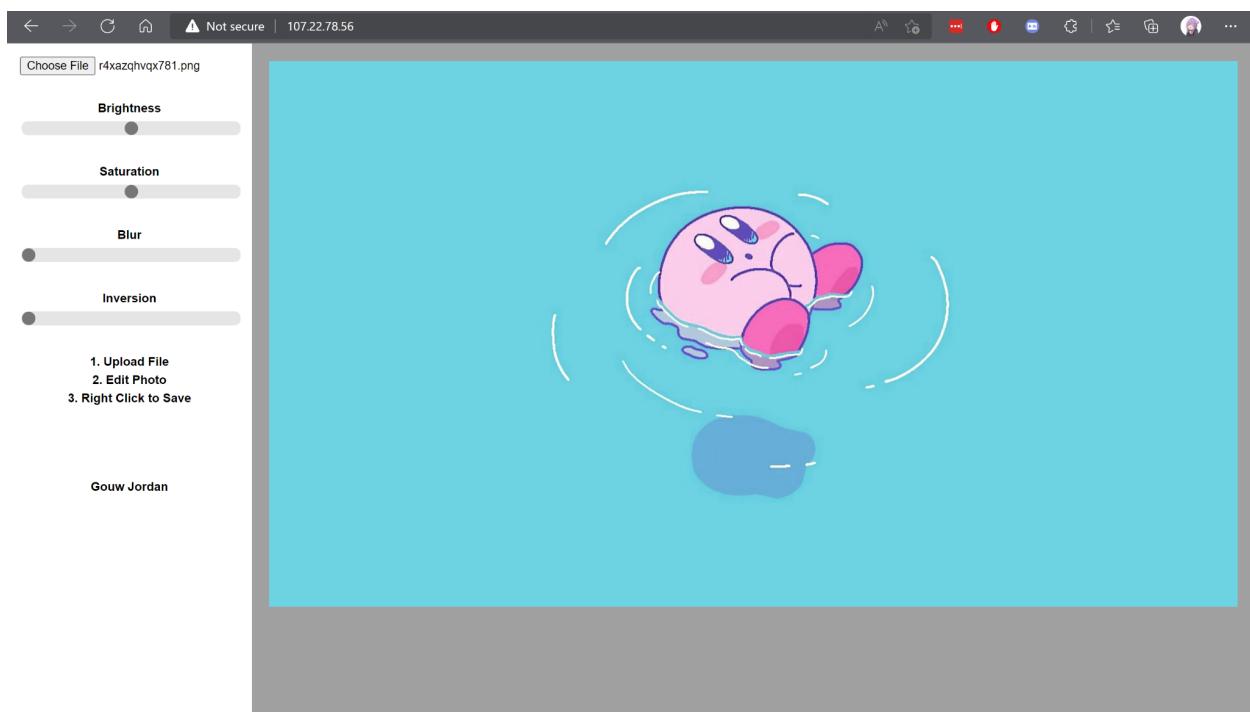
EC2 > Elastic IP addresses > 107.22.78.56

107.22.78.56

Actions ▾ Associate Elastic IP address

Summary			
Allocated IPv4 address 107.22.78.56	Type Public IP	Allocation ID eipalloc-0b152b5672930e2d8	Reverse DNS record -
Association ID eipassoc-0c016c18166fcc1f6	Scope VPC	Associated instance ID i-04205e6ac42c2c44b	Private IP address 172.31.6.214
Network interface ID eni-0eaf4abef0b3b20fd	Network interface owner account ID 894671075438	Public DNS ec2-107-22-78-56.compute-1.amazonaws.com	NAT Gateway ID -
Address pool Amazon	Network Border Group us-east-1		

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.

Amazon Machine Images (AMIs) (1) Info				
Owned by me		Search	Recycle Bin	EC2 Image Builder
Name	AMI ID	AMI name	Source	Owner
-	ami-05b2ebcba8cf788dd	photoeditor-image	894671075438/photoeditor-image	894671075438

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.

Name and tags [Info](#)

Name
photoeditorimage [Add additional tags](#)

▼ Application and OS Images (Amazon Machine Image) [Info](#)
An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

AMI from catalog [Recents](#) [My AMIs](#) [Quick Start](#)

Search our full catalog including 1000s of application and OS images

Amazon Machine Image (AMI)
photoeditor-image
ami-05b2ebcba8cf788dd

Published	Architecture	Virtualization	Root device type	ENI Enabled
2022-06-18T03:44:05.00Z	x86_64	hvm	ebs	Yes

Browse more AMIs
Including AMIs from AWS, Marketplace and the Community

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

Key pair (login) Info
You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - **required**
 Create new key pair

Network settings Edit

Network
vpc-096971d174395d9d0

Subnet
No preference (Default subnet in any availability zone)

Auto-assign public IP
Enable

Firewall (security groups) Info
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group Select existing security group

Common security groups Info
 Compare security group rules

photoeditor-vpc sg-008179d52bee3a809 X
VPC: vpc-096971d174395d9d0

Summary

Number of instances Info

Software Image (AMI)
created image photoeditor
ami-05b2ebcba8cf788dd

Virtual server type (instance type)
t2.small

Firewall (security group)
photoeditor-vpc

Storage (volumes)
1 volume(s) - 8 GiB

Cancel Launch instance

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

Instances (4) <small>Info</small>								<input type="button" value="Connect"/>	Instance state	Actions	Launch instances	▼
								<input type="button" value="Search"/>				
	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone					
<input type="checkbox"/>	photoeditorproject	i-04205e6ac42c2c44b	Running	OK	2/2 checks passed	No alarms	+ us-east-1d					
<input type="checkbox"/>	photoeditorimage	i-0b71f04f50075b309	Pending	OK	-	No alarms	+ us-east-1d					

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

Elastic IP addresses (1/2)					<input type="button" value="Actions"/>	Allocate Elastic IP address	◀ 1 ▶ ⚙
					<input type="button" value="Filter Elastic IP addresses"/>		
	Name	Allocated IPv4 add...	Type	Allocation ID		Reverse DNS record	
<input type="checkbox"/>	-	107.22.78.56	Public IP	eipalloc-0b152b5672930e2d8		-	
<input checked="" type="checkbox"/>	-	35.169.244.105	Public IP	eipalloc-08639e11a4a021ca9		-	

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

The screenshot shows the AWS Elastic Load Balancing (ELB) console. At the top, there is a search bar with the text "photoeditor-ELB" and a "Create Load Balancer" button. Below the search bar is a table with columns: Name, DNS name, State, VPC ID, Availability Zones, and Type. One row is visible: "photoeditor-ELB" with "photoeditor-ELB-2088268699" as the DNS name, "Provisioning" state, "vpc-096971d174395d9d0" VPC ID, "us-east-1c, us-east-1b, ..." Availability Zones, and "application" Type. Below the table, the text "Load balancer: photoeditor-ELB" is displayed. Underneath, there are tabs for Description, Listeners, Monitoring, Integrated services, and Tags, with "Description" being the active tab. The "Basic Configuration" section contains the following details:

Name	photoeditor-ELB
ARN	arn:aws:elasticloadbalancing:us-east-1:894671075438:loadbalancer/app/photoeditor-ELB/bf20b74d3f6a308b
DNS name	photoeditor-ELB-2088268699.us-east-1.elb.amazonaws.com (A Record)
State	Provisioning
Type	application
Scheme	internet-facing
IP address type	ipv4
Edit IP address type	
VPC	vpc-096971d174395d9d0

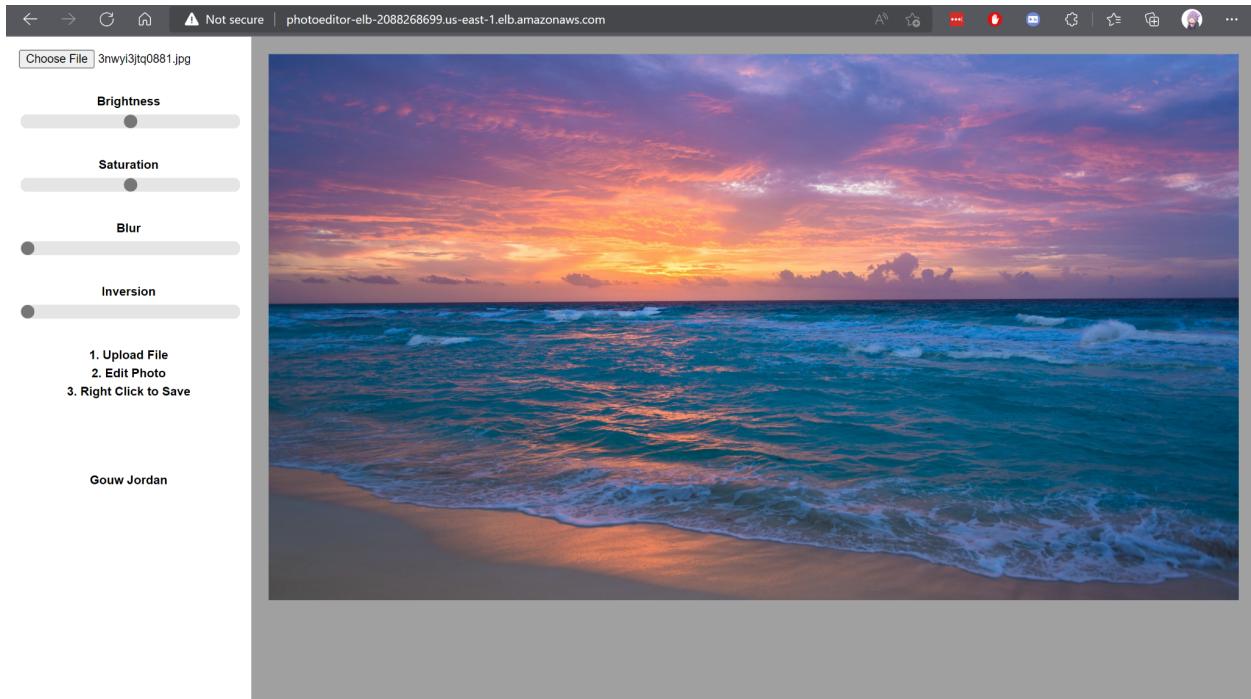
And I have created a target group and registered my two instances to the target group for the elb to distribute.

The screenshot shows the AWS EC2 Target groups console. At the top, there is a breadcrumb navigation "EC2 > Target groups" and a "Create target group" button. Below is a table titled "Target groups (1/1)" with columns: Name, ARN, Port, Protocol, Target type, and Load balancer. One row is visible: "test" with "arn:aws:elasticloadbalancing:us-east-1:894671075438:targetgroup/test/1234567890123456" as the ARN, port 80, protocol HTTP, target type Instance, and load balancer photoeditor-ELB. Below the table, the text "Target group: test" is displayed. Underneath, there is a section titled "Registered targets (2)" with a table showing two entries:

Instance ID	Name	Port	Zone	Health status	Health status details
i-008bbaecce6996cbf	photoeditor_image	80	us-east-1a	healthy	(green checkmark)
i-04205e6ac42c2c44b	photoeditorproject	80	us-east-1d	healthy	(green checkmark)

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...!