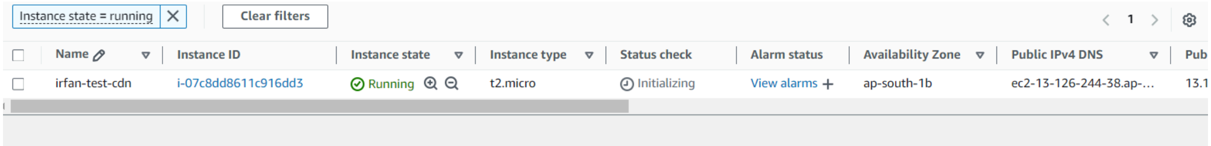
CLOUDFRONT

Amazon CloudFront is a **content delivery network (CDN)** service that securely delivers data, videos, applications, and APIs to users globally with low latency and high transfer speeds. It uses a global network of edge locations to cache content closer to the users for faster access.

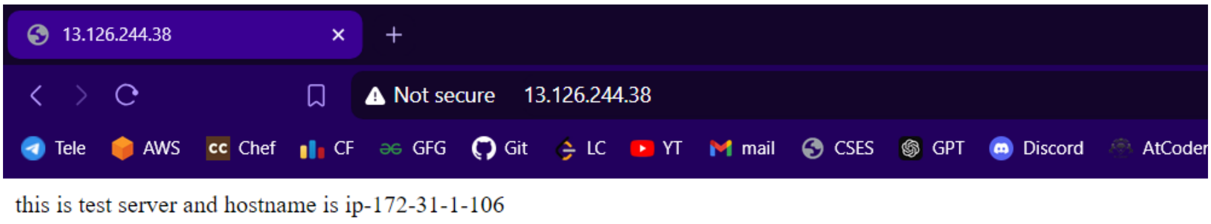
Key Features of CloudFront:

1. **Global Edge Locations**: CloudFront caches content at multiple edge locations worldwide, reducing the distance between users and the data, improving load times.
2. **Security**: It integrates with AWS Shield for DDoS protection, AWS Web Application Firewall (WAF) for application layer security, and supports HTTPS for secure data transfer.
3. **Dynamic and Static Content**: It serves both static assets (images, videos) and dynamic content (APIs, web applications) efficiently.
4. **Customizable**: You can customise how content is cached and delivered using Lambda@Edge, allowing you to run code at AWS edge locations.
5. **Use Cases**: Ideal for websites, APIs, video streaming, and accelerating the delivery of any content to users globally.

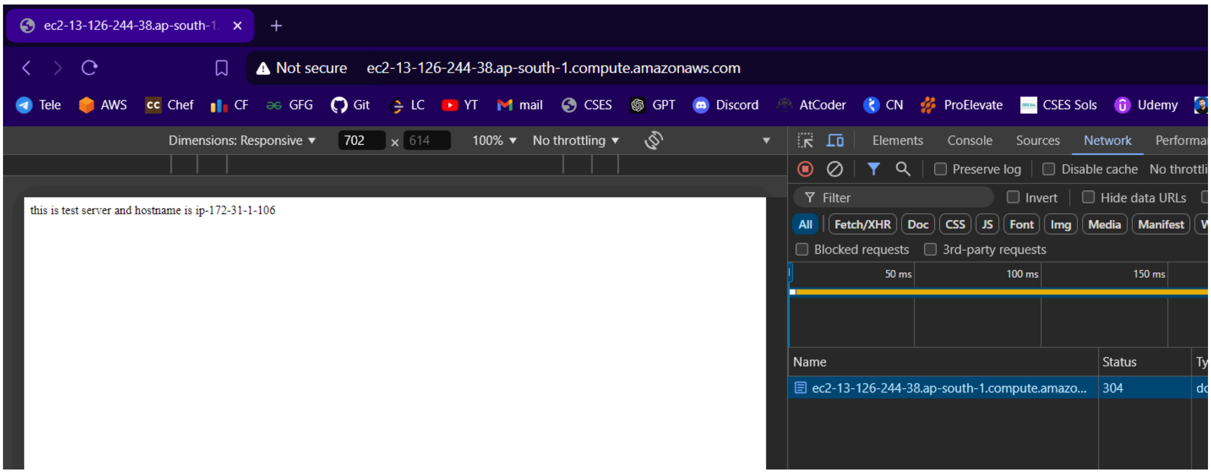
Create an EC2 Instance.



□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Pub
□	irfan-test-cdn	i-07c8dd8611c916dd3	Running	t2.micro	Initializing	View alarms +	ap-south-1b	ec2-13-126-244-38.ap-...	13.1



this is test server and hostname is ip-172-31-1-106

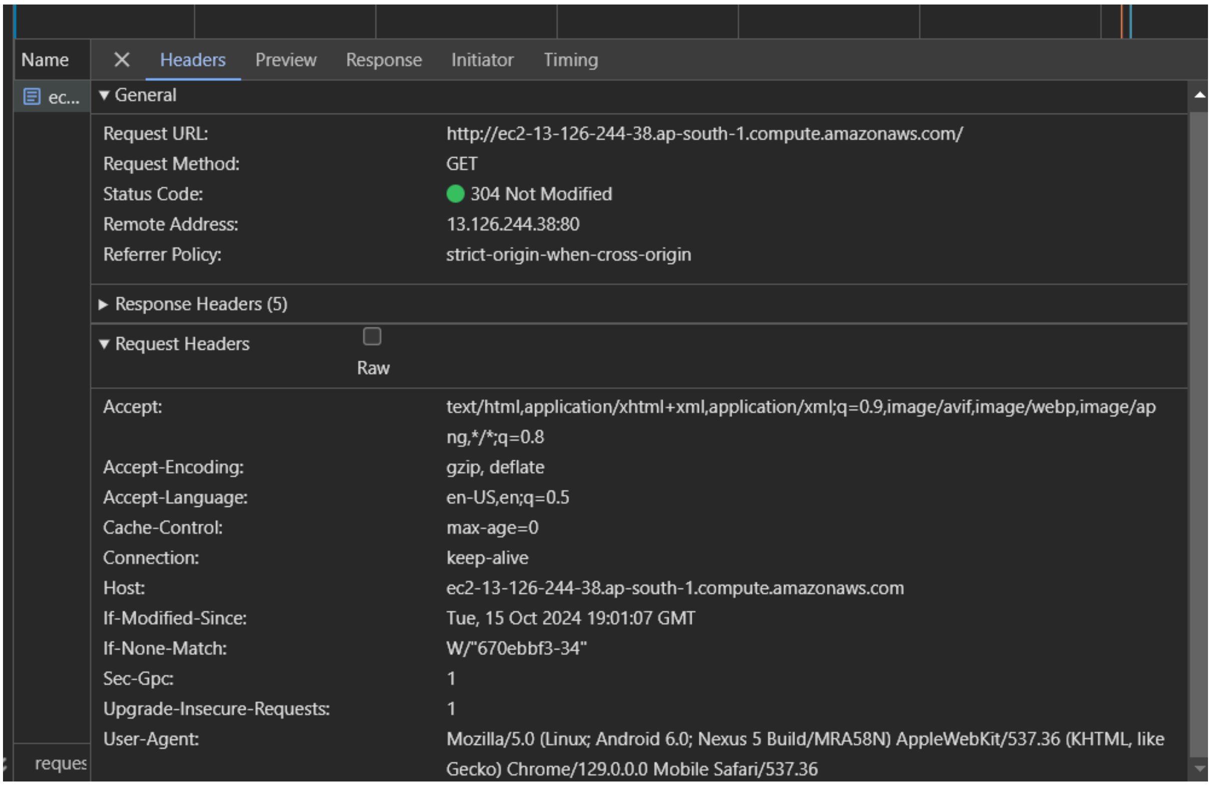


ec2-13-126-244-38.ap-south-1
Not secure
Tele AWS Chef CF GFG Git LC YT mail CSES GPT Discord AtCoder CN ProElevate CSES Sols Udemy
Dimensions: Responsive 702 × 614 100% No throttling
Elements Console Sources Network Performa
Preserve log Disable cache No throttli
Filter Invert Hide data URLs
All Fetch/XHR Doc CSS JS Font Img Media Manifest
Blocked requests 3rd-party requests
50 ms 100 ms 150 ms

ec2-13-126-244-38.ap-south-1.compute.amazonaws.com

this is test server and hostname is ip-172-31-1-106

Name	Status
ec2-13-126-244-38.ap-south-1.compute.amazo...	304



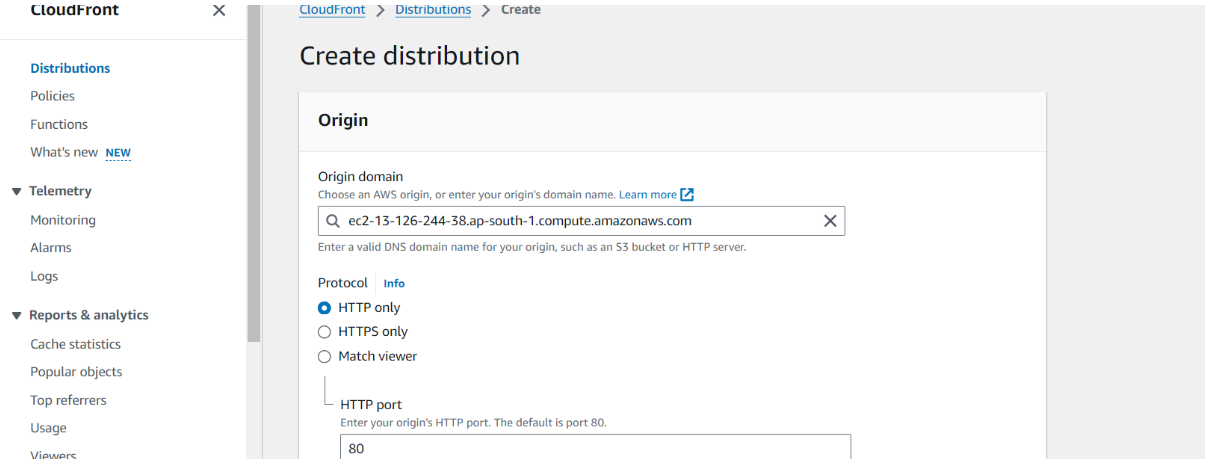
▼ General

Request URL: http://ec2-13-126-244-38.ap-south-1.compute.amazonaws.com/
Request Method: GET
Status Code: ● 304 Not Modified
Remote Address: 13.126.244.38:80
Referrer Policy: strict-origin-when-cross-origin

▶ Response Headers (5)

▼ Request Headers □ Raw

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.5
Cache-Control: max-age=0
Connection: keep-alive
Host: ec2-13-126-244-38.ap-south-1.compute.amazonaws.com
If-Modified-Since: Tue, 15 Oct 2024 19:01:07 GMT
If-None-Match: W/"670ebbf3-34"
Sec-Gpc: 1
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Linux; Android 6.0; Nexus 5 Build/MRA58N) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/129.0.0.0 Mobile Safari/537.36



Create distribution

Origin

Origin domain

Choose an AWS origin, or enter your origin's domain name. Learn more

ec2-13-126-244-38.ap-south-1.compute.amazonaws.com

Enter a valid DNS domain name for your origin, such as an S3 bucket or HTTP server.

Protocol Info

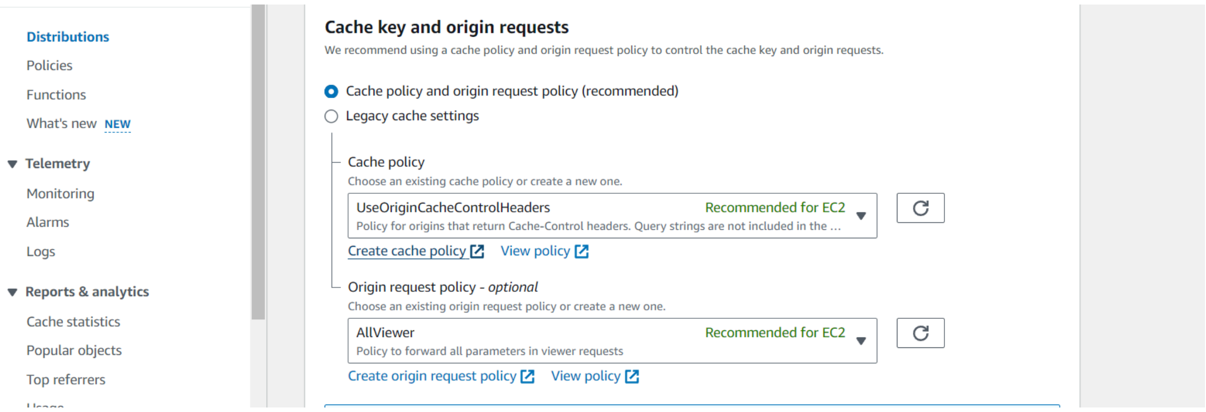
● HTTP only
○ HTTPS only
○ Match viewer

HTTP port

Enter your origin's HTTP port. The default is port 80.

80

Create a cloudfront Distribution.



Cache key and origin requests

We recommend using a cache policy and origin request policy to control the cache key and origin requests.

● Cache policy and origin request policy (recommended)
○ Legacy cache settings

Cache policy

Choose an existing cache policy or create a new one.

UseOriginCacheControlHeaders Recommended for EC2

Policy for origins that return Cache-Control headers. Query strings are not included in the ...

Create cache policy View policy

Origin request policy - *optional*

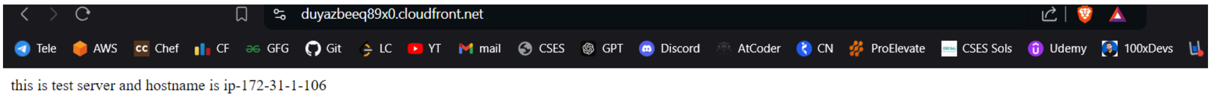
Choose an existing origin request policy or create a new one.

AllViewer Recommended for EC2

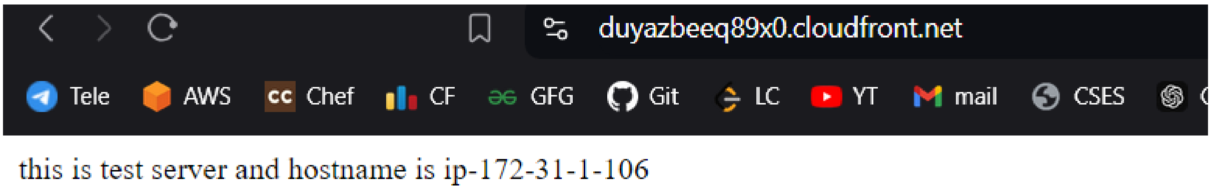
Policy to forward all parameters in viewer requests

Create origin request policy View policy

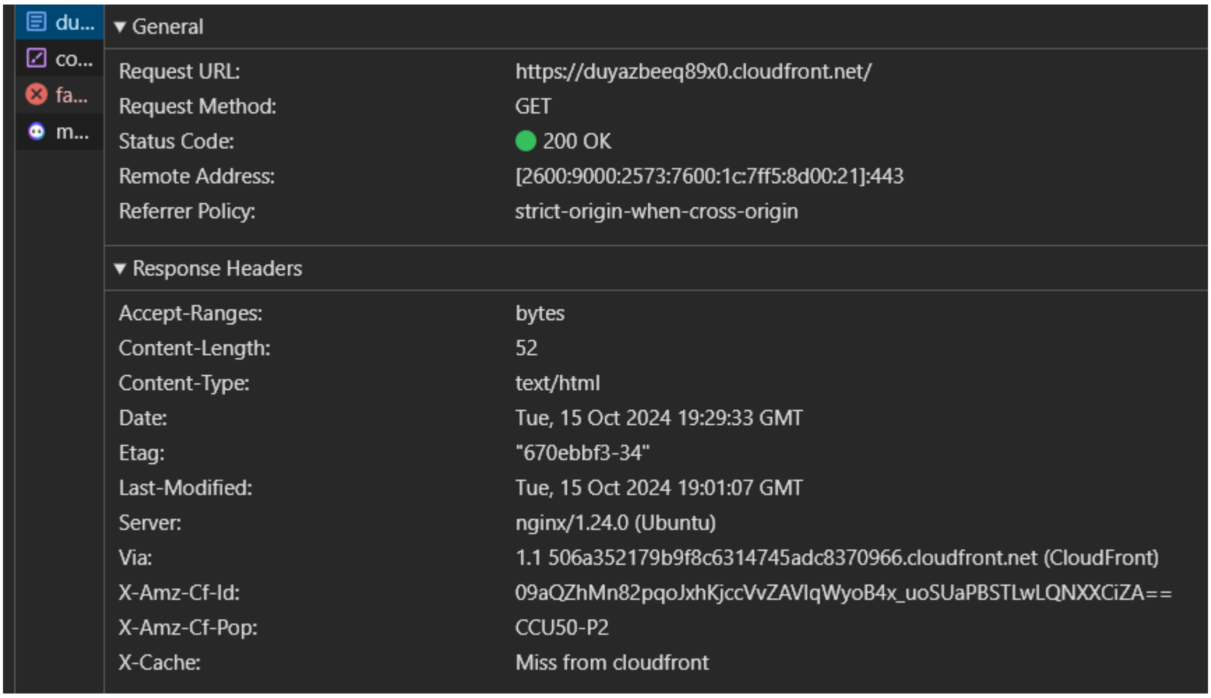
Create a Cache Policy.



this is test server and hostname is ip-172-31-1-106



this is test server and hostname is ip-172-31-1-106

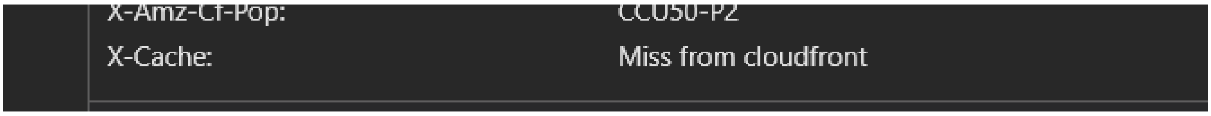


General

Request URL: https://duyazbeeq89x0.cloudfront.net/
Request Method: GET
Status Code: ● 200 OK
Remote Address: [2600:9000:2573:7600:1c:7ff5:8d00:21]:443
Referrer Policy: strict-origin-when-cross-origin

Response Headers

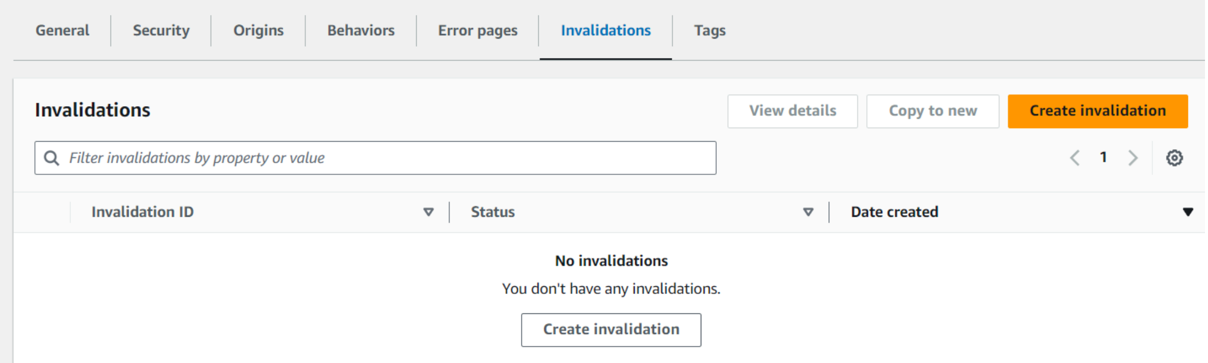
Accept-Ranges: bytes
Content-Length: 52
Content-Type: text/html
Date: Tue, 15 Oct 2024 19:29:33 GMT
Etag: "670ebbf3-34"
Last-Modified: Tue, 15 Oct 2024 19:01:07 GMT
Server: nginx/1.24.0 (Ubuntu)
Via: 1.1 506a352179b9f8c6314745adc8370966.cloudfront.net (CloudFront)
X-Amz-Cf-Id: 09aQZhMn82pqoJxhKjccVvZAVIqWyoB4x_uoSUaPBSTLwLQNXXCiZA==
X-Amz-Cf-Pop: CCU50-P2
X-Cache: Miss from cloudfront


X-Amz-Cf-Pop: CCU50-P2
X-Cache: Miss from cloudfront

Fetched from CDN.

If we now make changes in the homepage. The changes wil not reflect in the Distribution Domain. As previous cached data will be shown.
Distribution domain name
duyazbeeq89x0.cloudfront.net

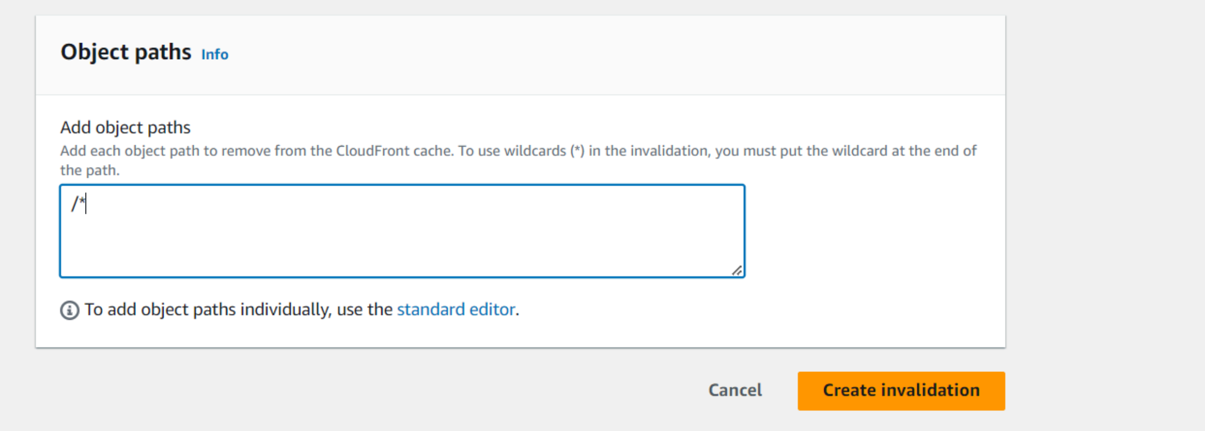
To solve this . Create Invalidation.



Invalidations

Invalidation ID	Status	Date created

No invalidations
You don't have any invalidations.



Object paths Info

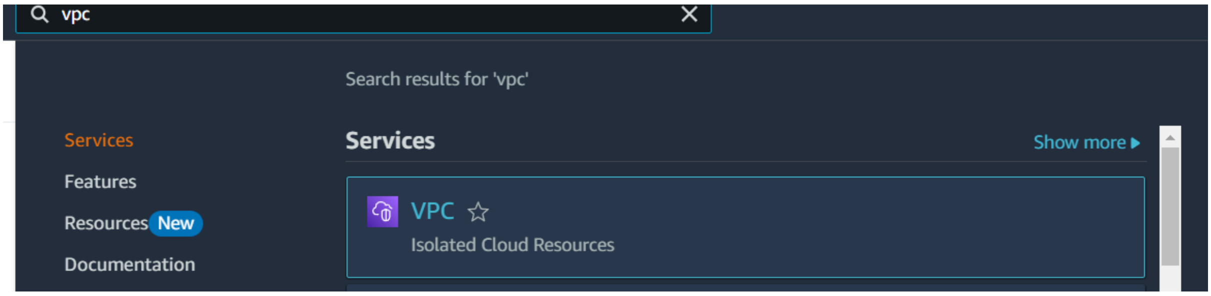
Add object paths

Add each object path to remove from the CloudFront cache. To use wildcards (*) in the invalidation, you must put the wildcard at the end of the path.

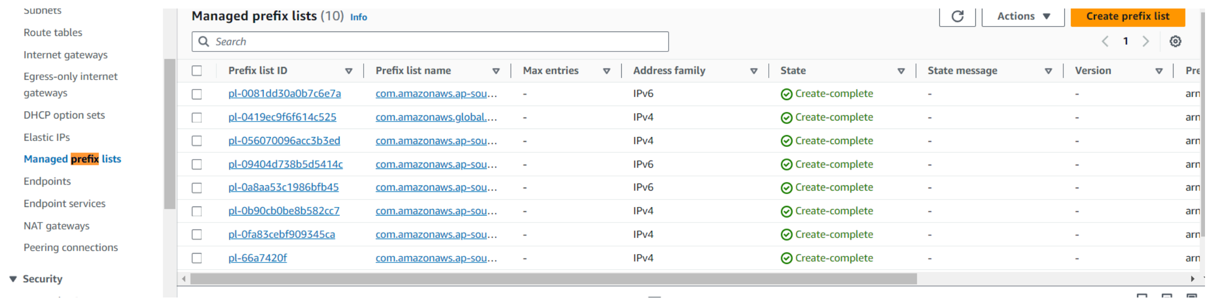
```
/*
```

To add object paths individually, use the standard editor.

How to make EC2/ALB Instance Accessible from CloudFront Only

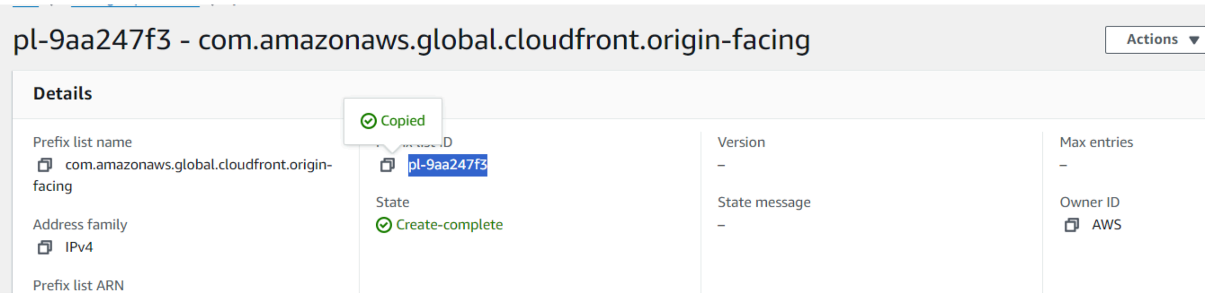


Go to VPS.



Managed prefix lists (10) Info

□	Prefix list ID	Prefix list name	Max entries	Address family	State	State message	Version	Pre
□	pl-0081dd30a0b7c6e7a	com.amazonaws.ap-sou…	-	IPv6	✓ Create-complete	-	-	arn
□	pl-0419ec9f6f614c525	com.amazonaws.global.…	-	IPv4	✓ Create-complete	-	-	arn
□	pl-056070096acc3b3ed	com.amazonaws.ap-sou…	-	IPv4	✓ Create-complete	-	-	arn
□	pl-09404d738b5d5414c	com.amazonaws.ap-sou…	-	IPv6	✓ Create-complete	-	-	arn
□	pl-0a8aa53c1986bfb45	com.amazonaws.ap-sou…	-	IPv6	✓ Create-complete	-	-	arn
□	pl-0b90cb0be8b582cc7	com.amazonaws.ap-sou…	-	IPv4	✓ Create-complete	-	-	arn
□	pl-0fa83cebf909345ca	com.amazonaws.ap-sou…	-	IPv4	✓ Create-complete	-	-	arn
□	pl-66a7420f	com.amazonaws.ap-sou…	-	IPv4	✓ Create-complete	-	-	arn

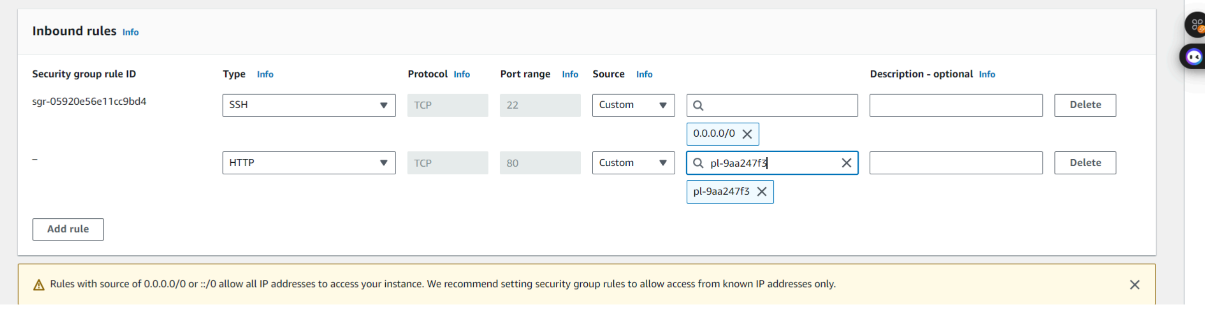


pl-9aa247f3 - com.amazonaws.global.cloudfront.origin-facing

Details

Prefix list name com.amazonaws.global.cloudfront.origin-facing	Prefix list ID pl-9aa247f3	Version –	Max entries –
Address family IPv4	State ✓ Create-complete	State message –	Owner ID AWS
Prefix list ARN			

Select Security Group. Select Inbound rules to edit.

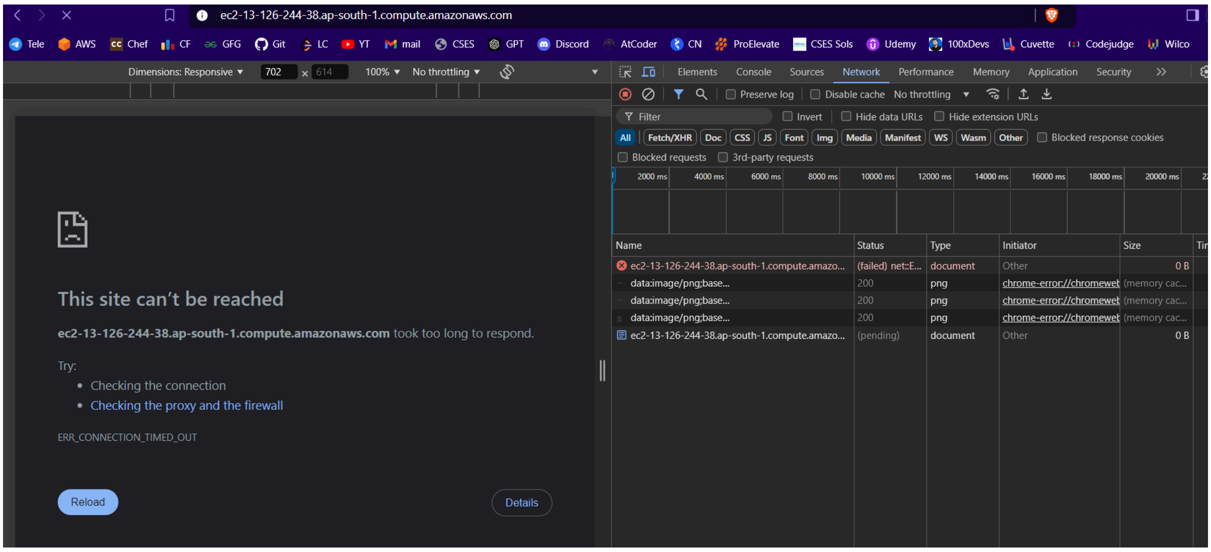


Inbound rules Info

Security group rule ID	Type	Protocol	Port range	Source		Description - optional	
sgr-05920e56e11cc9bd4	SSH	TCP	22	Custom	0.0.0.0/0		Delete
–	HTTP	TCP	80	Custom	pl-9aa247f3		Delete

Add rule

⚠ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.



This site can't be reached

ec2-13-126-244-38.ap-south-1.compute.amazonaws.com took too long to respond.

Try:

- Checking the connection
- Checking the proxy and the firewall

ERR_CONNECTION_TIMED_OUT

Reload

Details

Name	Status	Type	Initiator	Size
ec2-13-126-244-38.ap-south-1.compute.amazo...	(failed) net::E...	document	Other	0 B
data:image/png;base...	200	png	chrome-error://chromewet	(memory cac...
data:image/png;base...	200	png	chrome-error://chromewet	(memory cac...
data:image/png;base...	200	png	chrome-error://chromewet	(memory cac...
ec2-13-126-244-38.ap-south-1.compute.amazo...	(pending)	document	Other	0 B

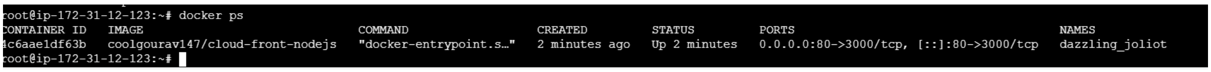
Project

Installed Docker on my EC2.
Pulled an image of nodejs Application.

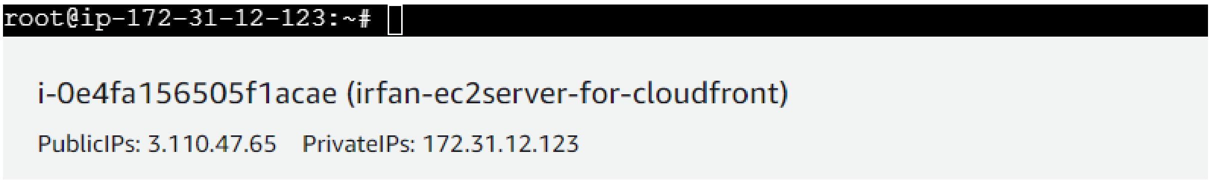
```
docker run -itd -p 80:3000 {name}
```

Breakdown of the command:

- `docker run`: Runs a new container from an image.
- `-i`: Interactive mode (keeps STDIN open).
- `-t`: Allocates a pseudo-TTY (terminal).
- `-d`: Runs the container in detached mode (in the background).
- `-p 80:3000`: Maps port 80 on the host (EC2 instance) to port 3000 in the container. This means the app running on port 3000 inside the container will be accessible on port 80 from outside.



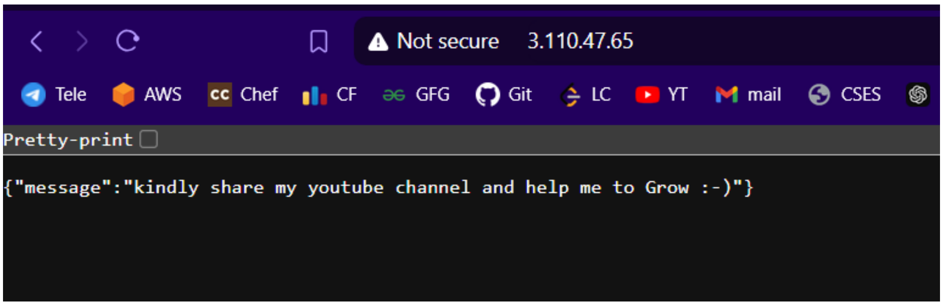
```
root@ip-172-31-12-123:~# docker ps
CONTAINER ID   IMAGE                          COMMAND                  CREATED         STATUS         PORTS                                 NAMES
4c6aae1df63b   coolgourav147/cloud-front-nodejs   "docker-entrypoint.s…"   2 minutes ago   Up 2 minutes   0.0.0.0:80->3000/tcp, [::]:80->3000/tcp   dazzling_joliot
root@ip-172-31-12-123:~#
```



```
root@ip-172-31-12-123:~#
```

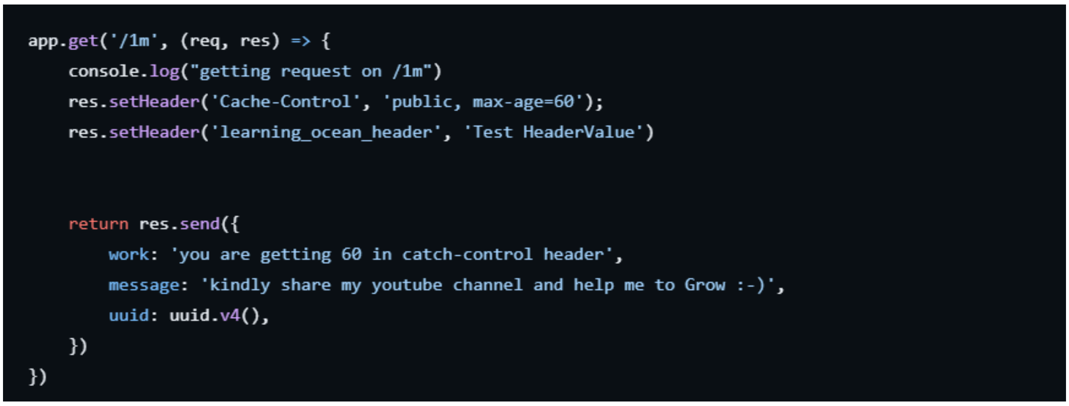
i-0e4fa156505f1acae (irfan-ec2server-for-cloudfront)

PublicIPs: 3.110.47.65   PrivateIPs: 172.31.12.123

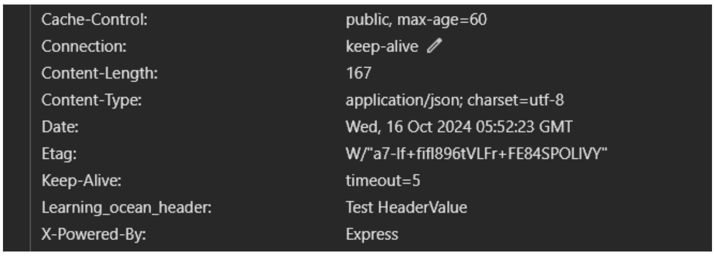


```
{"message":"kindly share my youtube channel and help me to Grow :-)"}
```

ROute /1m

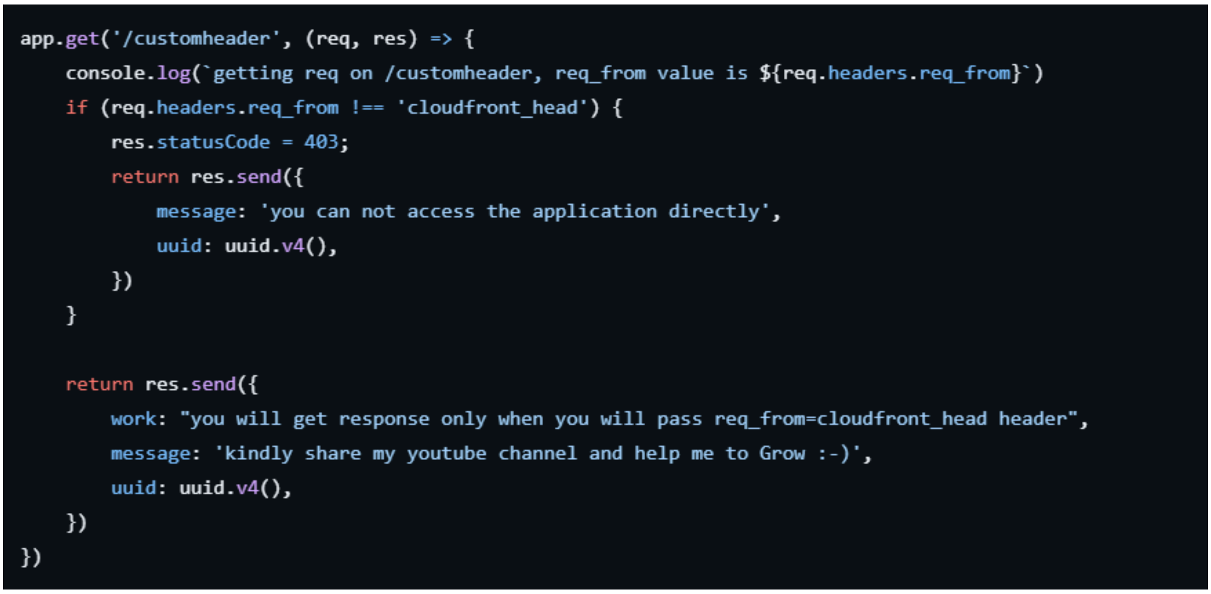


```
app.get('/1m', (req, res) => {
    console.log("getting request on /1m")
    res.setHeader('Cache-Control', 'public, max-age=60');
    res.setHeader('learning_ocean_header', 'Test HeaderValue')


    return res.send({
        work: 'you are getting 60 in catch-control header',
        message: 'kindly share my youtube channel and help me to Grow :-)',
        uuid: uuid.v4(),
    })
})
```



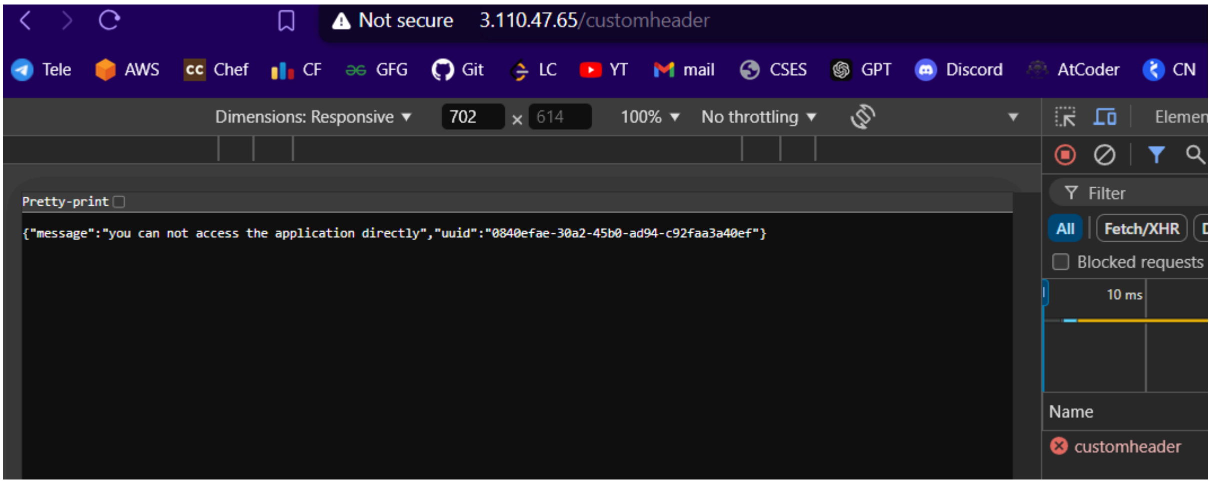
Cache-Control:	public, max-age=60
Connection:	keep-alive
Content-Length:	167
Content-Type:	application/json; charset=utf-8
Date:	Wed, 16 Oct 2024 05:52:23 GMT
Etag:	W/"a7-lf+fifl896tVLFr+FE84SPOLIVY"
Keep-Alive:	timeout=5
Learning_ocean_header:	Test HeaderValue
X-Powered-By:	Express

Cache Expire and Custom Header in Action



```
app.get('/customheader', (req, res) => {
    console.log(`getting req on /customheader, req_from value is ${req.headers.req_from}`)
    if (req.headers.req_from !== 'cloudfront_head') {
        res.statusCode = 403;
        return res.send({
            message: 'you can not access the application directly',
            uuid: uuid.v4(),
        })
    }

    return res.send({
        work: "you will get response only when you will pass req_from=cloudfront_head header",
        message: 'kindly share my youtube channel and help me to Grow :-)',
        uuid: uuid.v4(),
    })
})
```

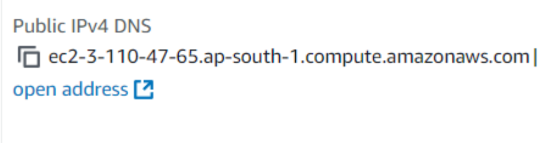
No header setup till now. So no response. 403.



```
{"message":"you can not access the application directly","uuid":"0840efae-30a2-45b0-ad94-c92faa3a40ef"}
```

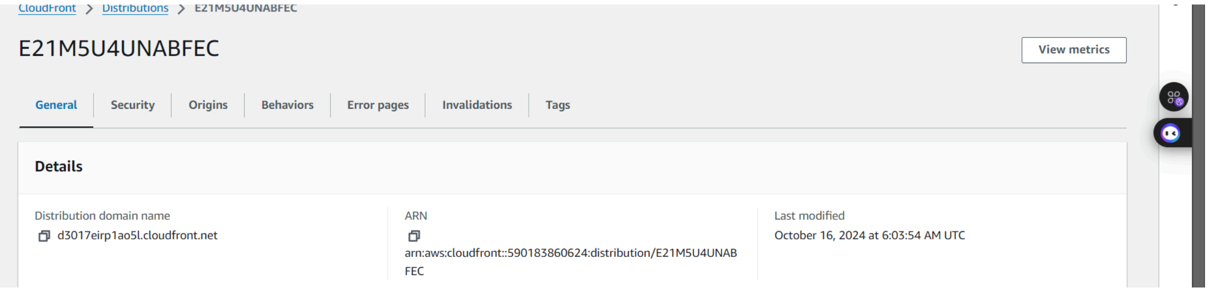
.

Create a Cloudfront distribution.



Public IPv4 DNS
ec2-3-110-47-65.ap-south-1.compute.amazonaws.com
open address

Copy it for Cloufront.



E21M5U4UNABFEC

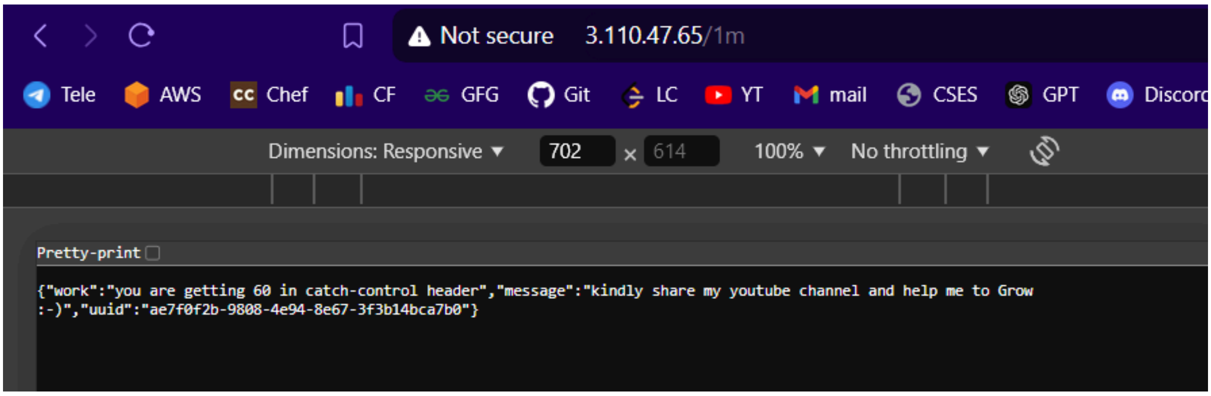
Details

Distribution domain name: d3017eirp1ao5l.cloudfront.net

ARN: arn:aws:cloudfront::590183860624:distribution/E21M5U4UNABFEC

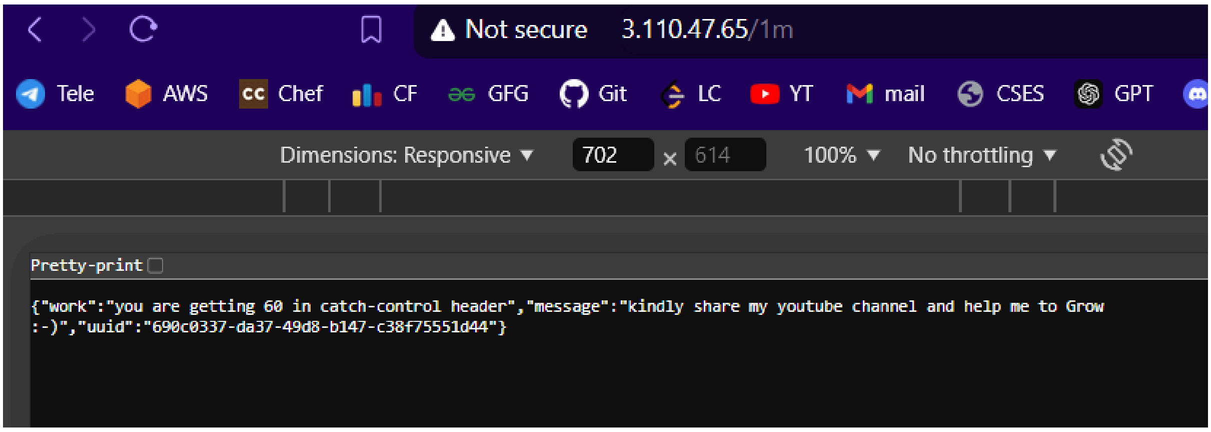
Last modified: October 16, 2024 at 6:03:54 AM UTC

Distribution Created.



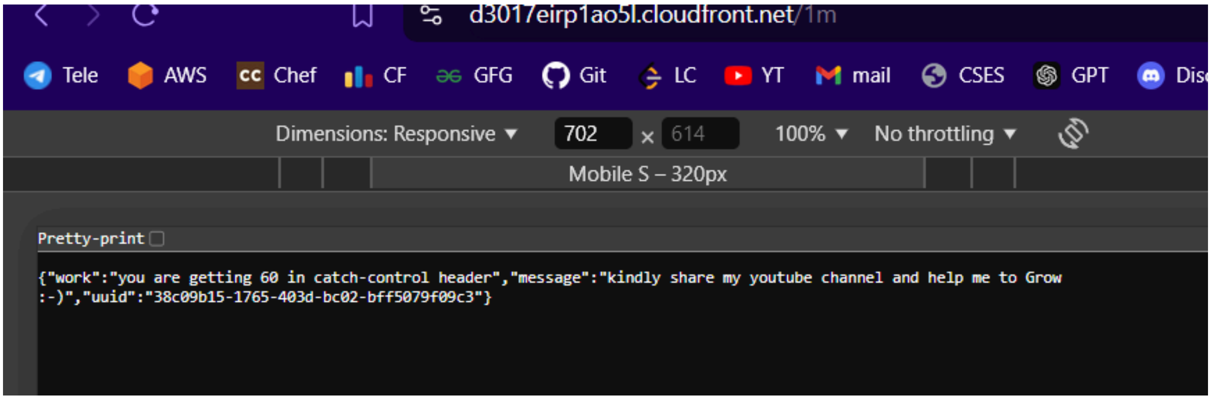
```
{"work":"you are getting 60 in catch-control header","message":"kindly share my youtube channel and help me to Grow
:-)","uuid":"ae7f0f2b-9808-4e94-8e67-3f3b14bca7b0"}
```

On refreshing.



```
{"work":"you are getting 60 in catch-control header","message":"kindly share my youtube channel and help me to Grow
:-)","uuid":"690c0337-da37-49d8-b147-c38f75551d44"}
```

UUID changes.

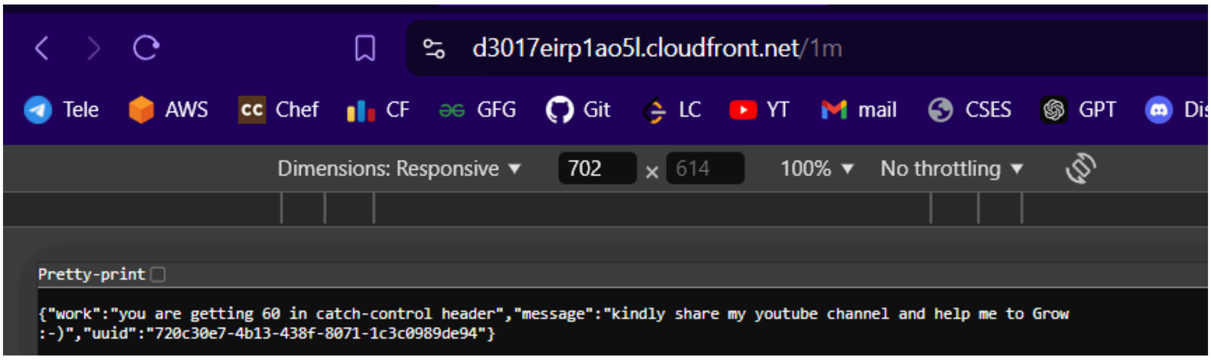


d3017eirp1ao5l.cloudfront.net/1m

Dimensions: Responsive ▾ 702 × 614 100% ▾ No throttling ▾

Pretty-print □

```
{"work":"you are getting 60 in catch-control header","message":"kindly share my youtube channel and help me to Grow
:-)","uuid":"38c09b15-1765-403d-bc02-bff5079f09c3"}
```

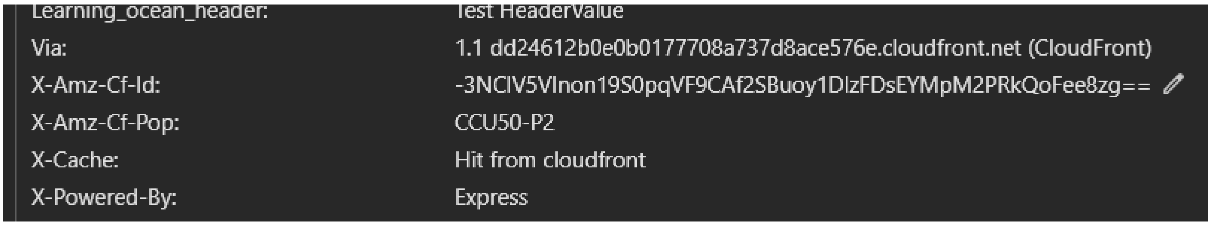


d3017eirp1ao5l.cloudfront.net/1m

Dimensions: Responsive ▾ 702 × 614 100% ▾ No throttling ▾

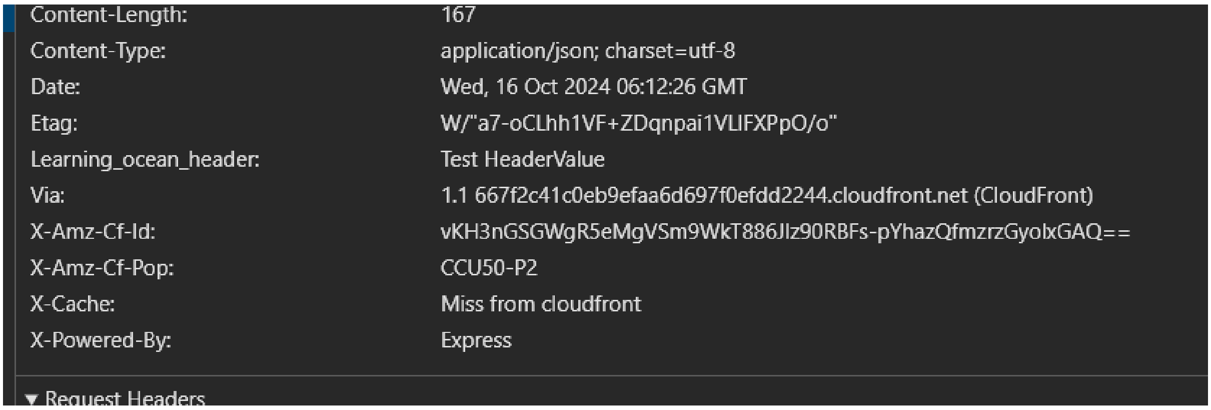
Pretty-print □

```
{"work":"you are getting 60 in catch-control header","message":"kindly share my youtube channel and help me to Grow
:-)","uuid":"720c30e7-4b13-438f-8071-1c3c0989de94"}
```



Learning_ocean_header:	Test HeaderValue
Via:	1.1 dd24612b0e0b0177708a737d8ace576e.cloudfront.net (CloudFront)
X-Amz-Cf-Id:	-3NCIV5VInon19S0pqVF9CAf2SBuoy1DlzFDsEYMpM2PRkQoFee8zg==
X-Amz-Cf-Pop:	CCU50-P2
X-Cache:	Hit from cloudfront
X-Powered-By:	Express

After 1 minute.

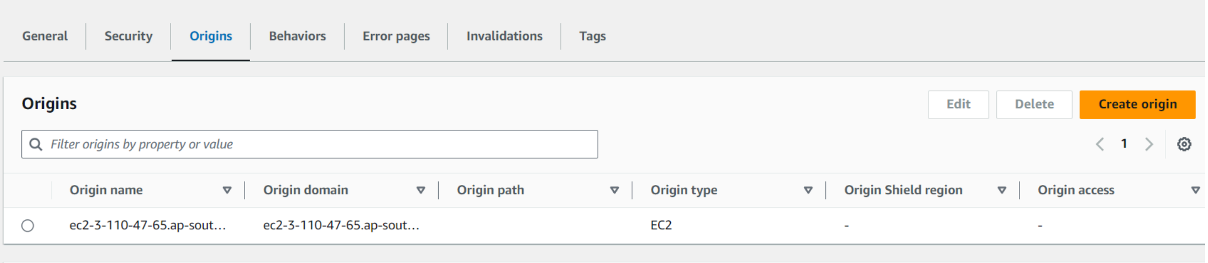


Content-Length:	167
Content-Type:	application/json; charset=utf-8
Date:	Wed, 16 Oct 2024 06:12:26 GMT
Etag:	W/"a7-oCLhh1VF+ZDqnpai1VLIFXPpO/o"
Learning_ocean_header:	Test HeaderValue
Via:	1.1 667f2c41c0eb9efaa6d697f0efdd2244.cloudfront.net (CloudFront)
X-Amz-Cf-Id:	vKH3nGSGWgR5eMgVSm9WkT886JIz90RBFs-pYhazQfmzrzGyoIxGAQ==
X-Amz-Cf-Pop:	CCU50-P2
X-Cache:	Miss from cloudfront
X-Powered-By:	Express

▾ Request Headers

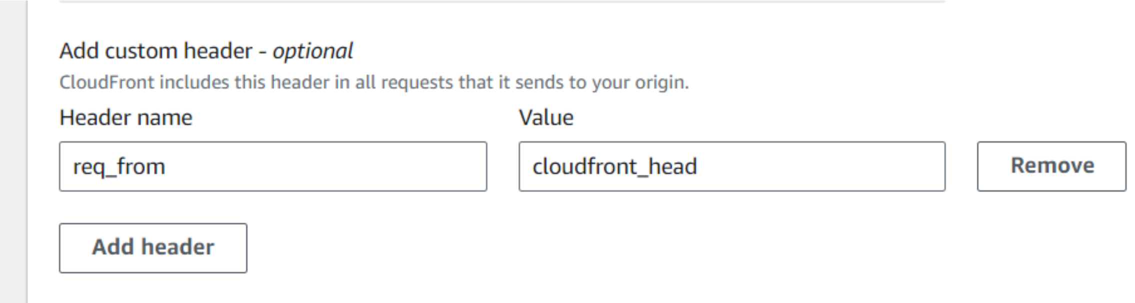
CDN working.

Edit Origin Of the Distribution.



Origins

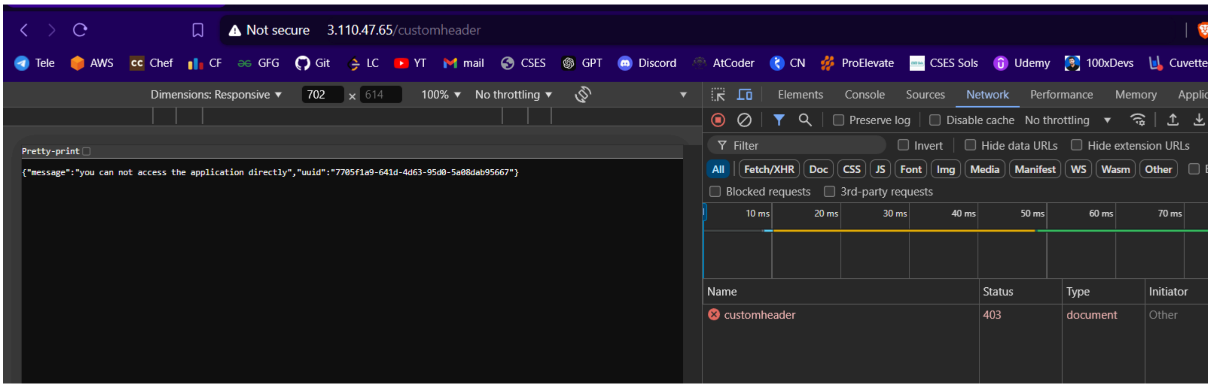
	Origin name	Origin domain	Origin path	Origin type	Origin Shield region	Origin access
○	ec2-3-110-47-65.ap-sout...	ec2-3-110-47-65.ap-sout...		EC2	-	-



Add custom header - *optional*

CloudFront includes this header in all requests that it sends to your origin.

Header name	Value
req_from	cloudfront_head



3.110.47.65/customheader

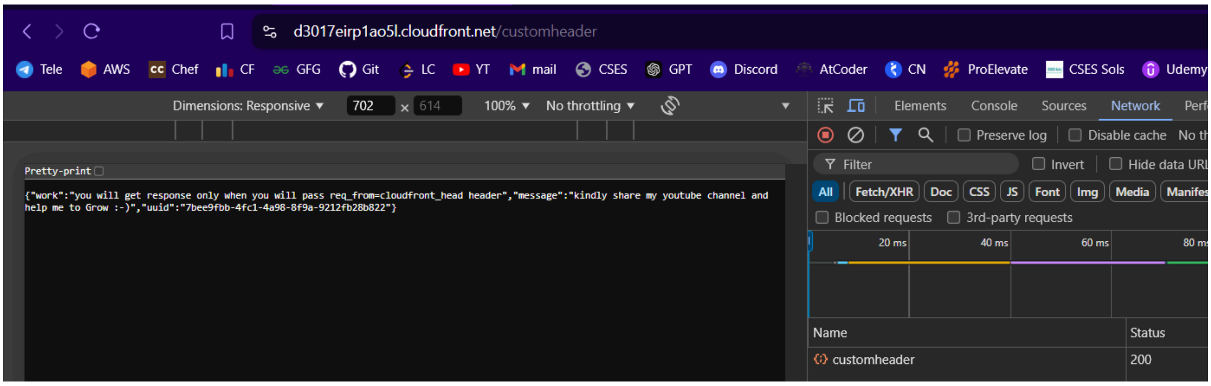
Pretty-print □

```
{"message":"you can not access the application directly","uuid":"7705f1a9-641d-4d63-95d0-5a08dab95667"}
```

Name	Status	Type	Initiator
customheader	403	document	Other

Not accessible from EC2.

Now that i have added the header in Cloudfront. It should be accessible from CLoudfront.



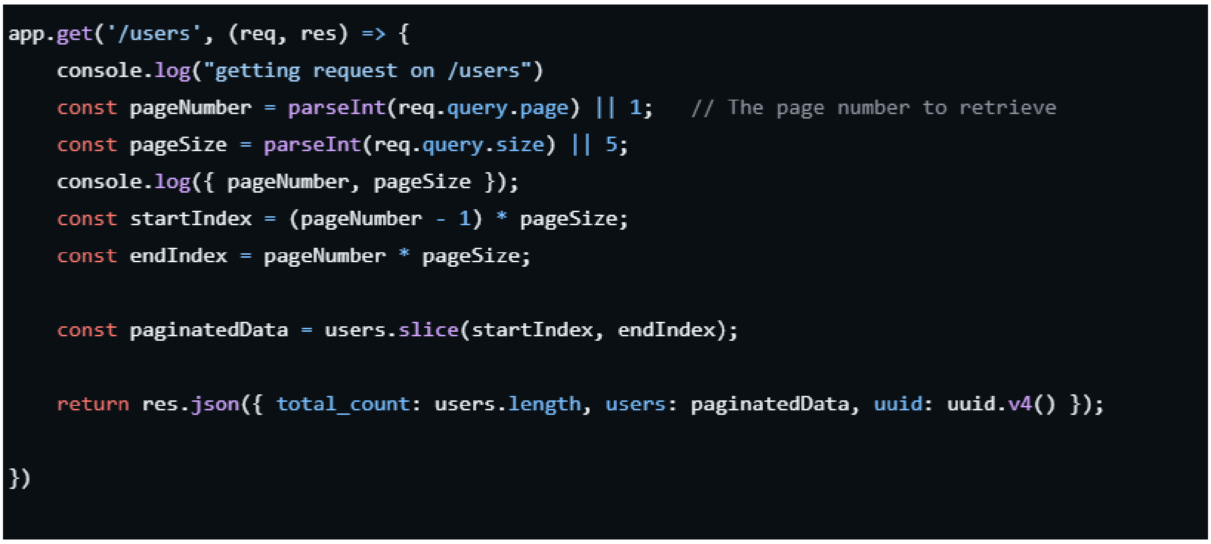
d3017eirp1ao5l.cloudfront.net/customheader

Pretty-print □

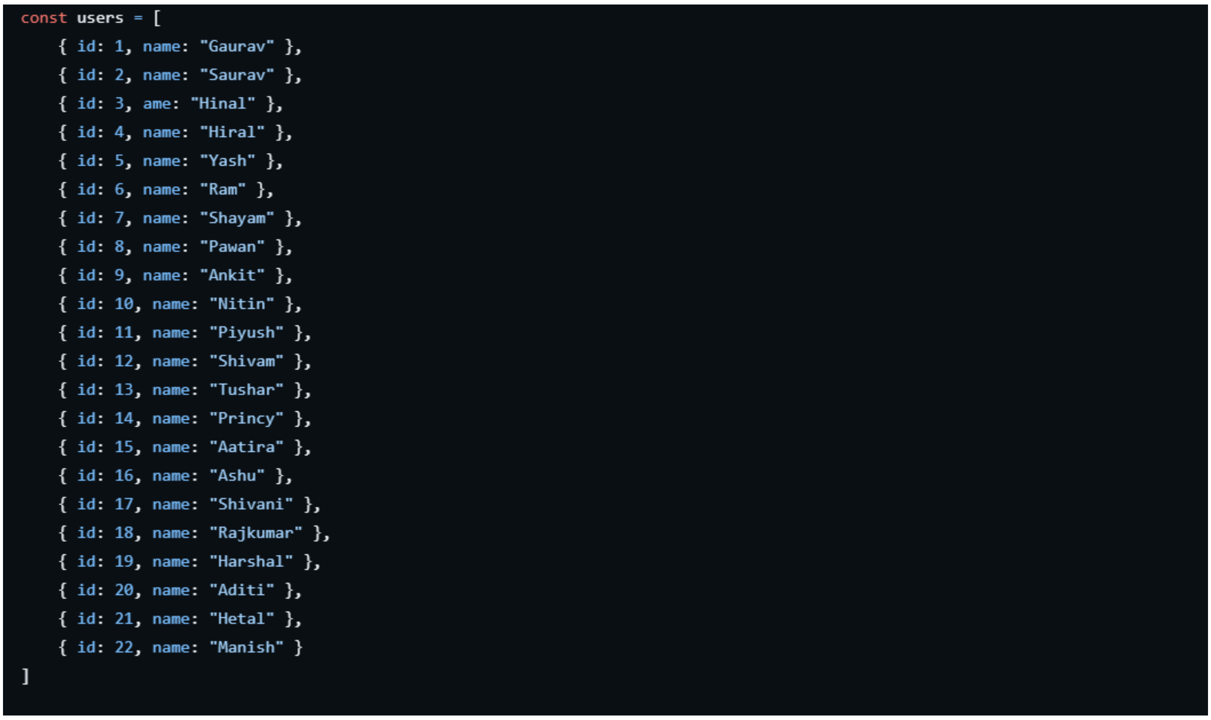
```
{"work":"you will get response only when you will pass req_from=cloudfront_head header","message":"kindly share my youtube channel and
help me to Grow :-)","uuid":"7bee9fbb-4fc1-4a98-8f9a-9212fb28b822"}
```

Name	Status
customheader	200

Testing Pagination.



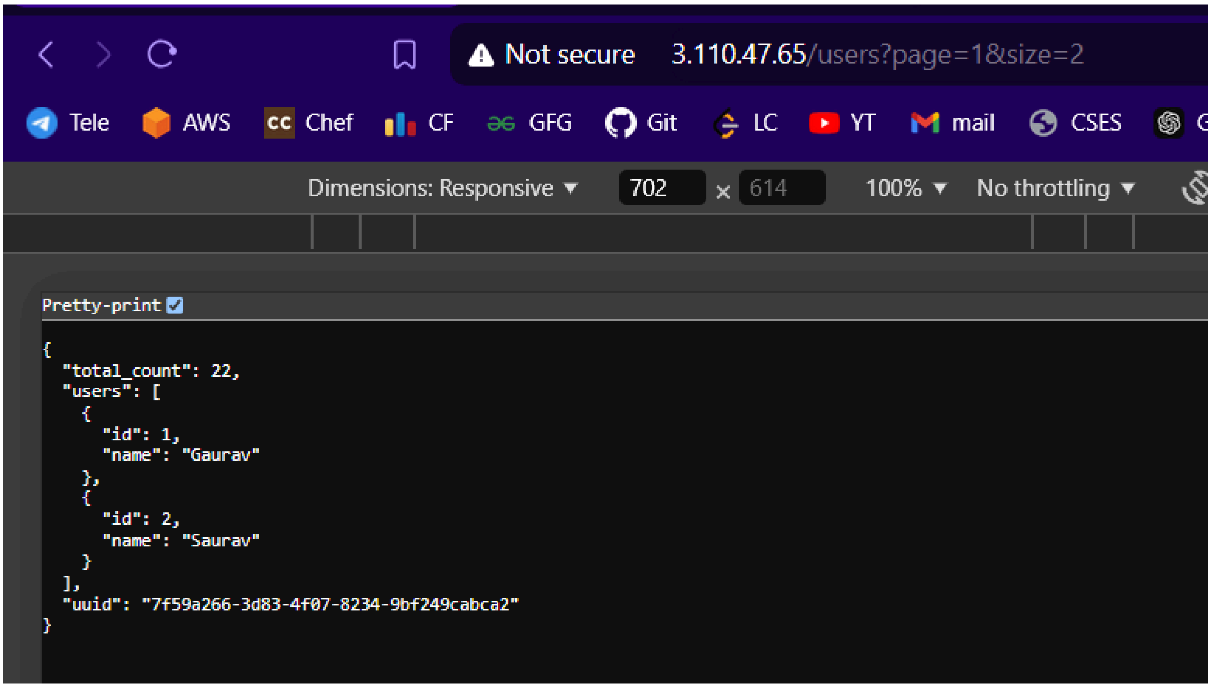
```
app.get('/users', (req, res) => {
    console.log("getting request on /users")
    const pageNumber = parseInt(req.query.page) || 1;   // The page number to retrieve
    const pageSize = parseInt(req.query.size) || 5;
    console.log({ pageNumber, pageSize });
    const startIndex = (pageNumber - 1) * pageSize;
    const endIndex = pageNumber * pageSize;

    const paginatedData = users.slice(startIndex, endIndex);

    return res.json({ total_count: users.length, users: paginatedData, uuid: uuid.v4() });

})
```



```
const users = [
    { id: 1, name: "Gaurav" },
    { id: 2, name: "Saurav" },
    { id: 3, ame: "Hinal" },
    { id: 4, name: "Hiral" },
    { id: 5, name: "Yash" },
    { id: 6, name: "Ram" },
    { id: 7, name: "Shayam" },
    { id: 8, name: "Pawan" },
    { id: 9, name: "Ankit" },
    { id: 10, name: "Nitin" },
    { id: 11, name: "Piyush" },
    { id: 12, name: "Shivam" },
    { id: 13, name: "Tushar" },
    { id: 14, name: "Princy" },
    { id: 15, name: "Aatira" },
    { id: 16, name: "Ashu" },
    { id: 17, name: "Shivani" },
    { id: 18, name: "Rajkumar" },
    { id: 19, name: "Harshal" },
    { id: 20, name: "Aditi" },
    { id: 21, name: "Hetal" },
    { id: 22, name: "Manish" }
]
```

Total 22 Users.

From Cloud front:

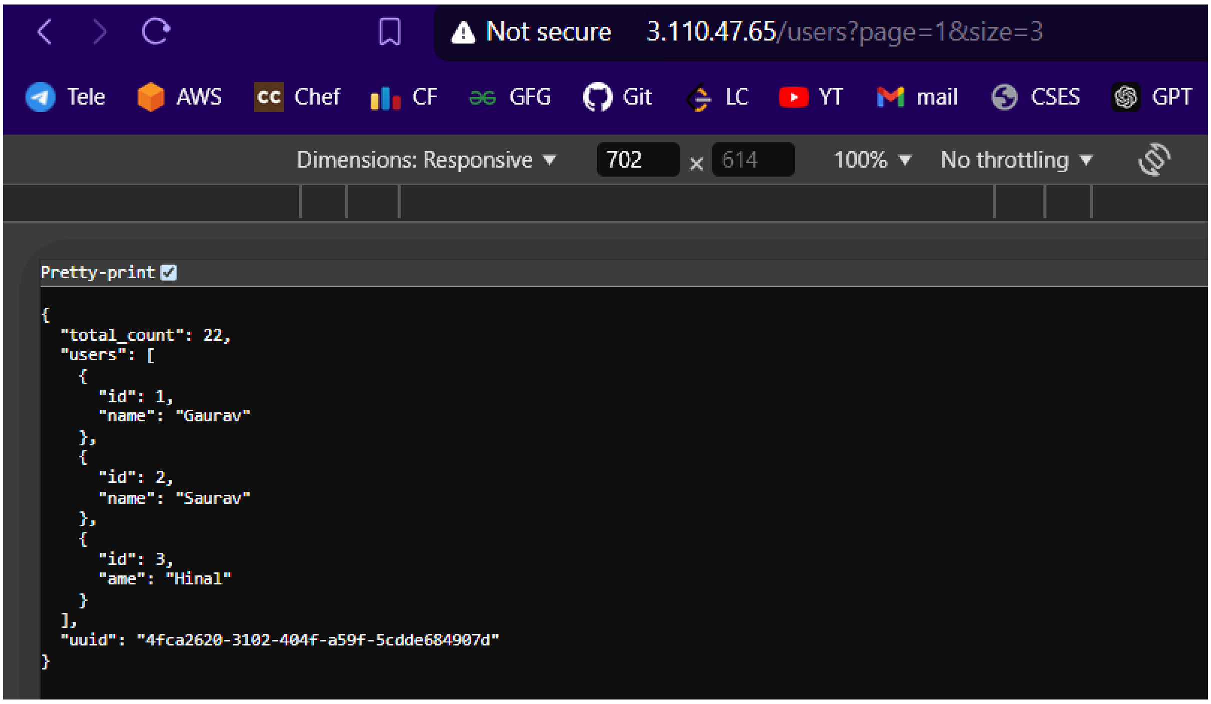


Not secure 3.110.47.65/users?page=1&size=2

Dimensions: Responsive ▾ 702 × 614 100% ▾ No throttling ▾

```
{
  "total_count": 22,
  "users": [
    {
      "id": 1,
      "name": "Gaurav"
    },
    {
      "id": 2,
      "name": "Saurav"
    }
  ],
  "uuid": "7f59a266-3d83-4f07-8234-9bf249cabca2"
}
```

size=2.



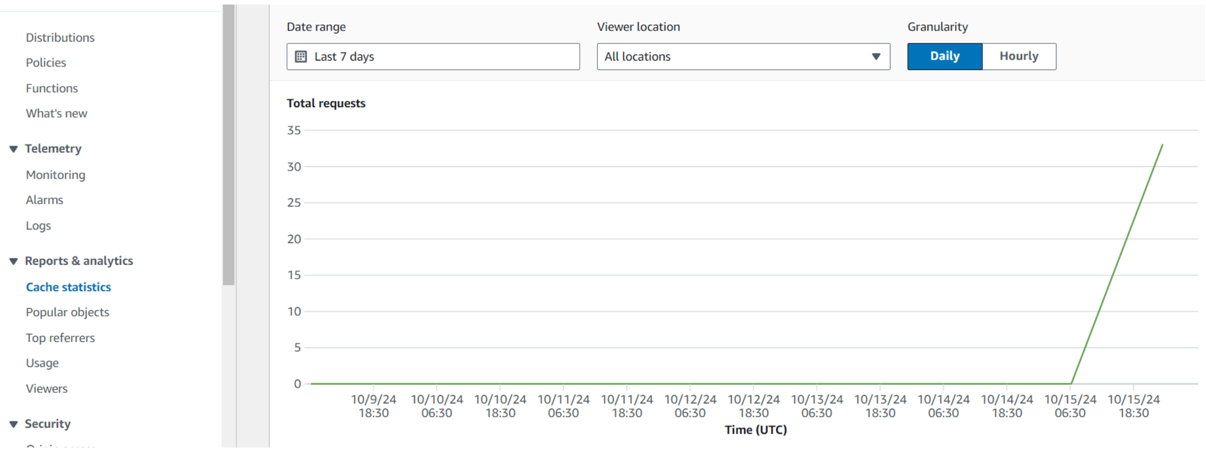
Not secure 3.110.47.65/users?page=1&size=3

Dimensions: Responsive ▾ 702 × 614 100% ▾ No throttling ▾

```
{
  "total_count": 22,
  "users": [
    {
      "id": 1,
      "name": "Gaurav"
    },
    {
      "id": 2,
      "name": "Saurav"
    },
    {
      "id": 3,
      "ame": "Hinal"
    }
  ],
  "uuid": "4fca2620-3102-404f-a59f-5cdde684907d"
}
```

SIze=3.

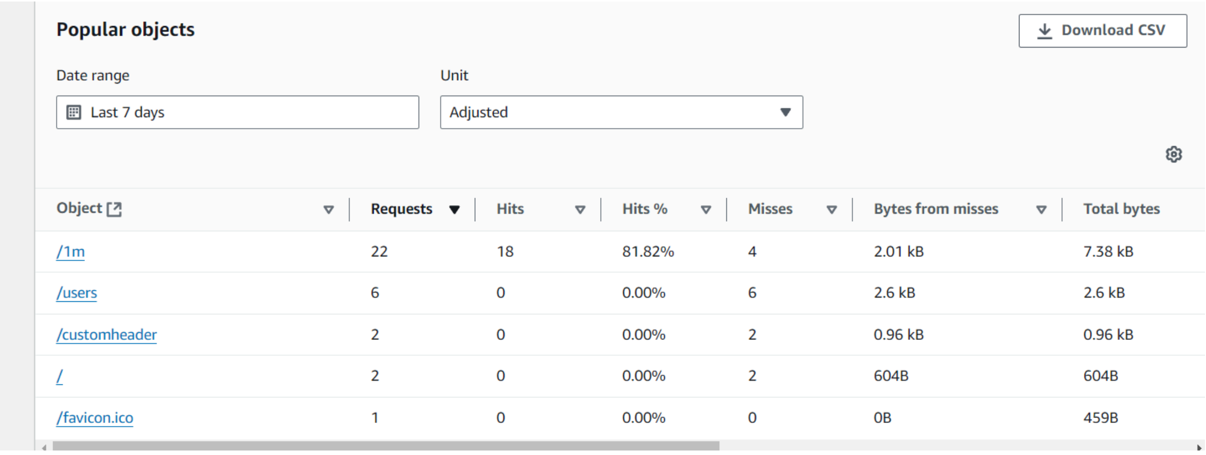
Statistics:



Date range: Last 7 days

Viewer location: All locations

Granularity: Daily | Hourly

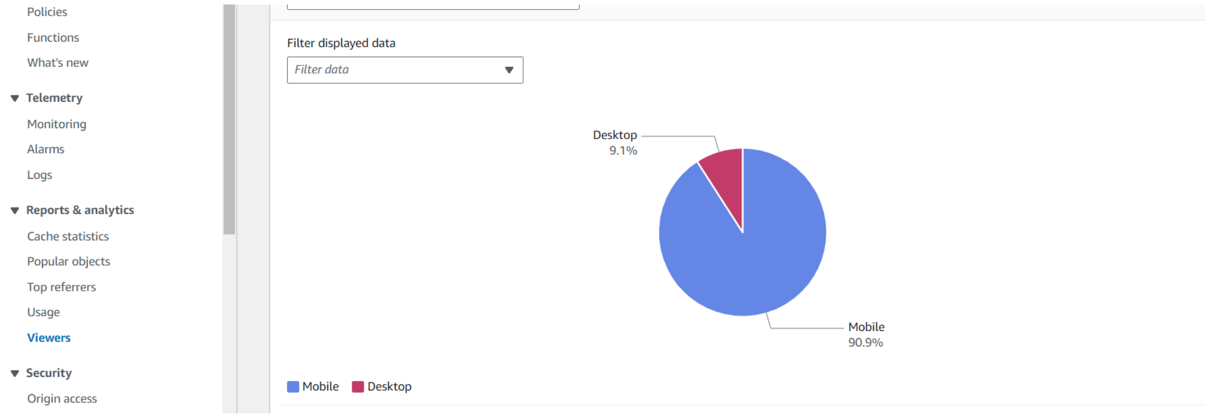


Popular objects

Date range: Last 7 days

Unit: Adjusted

Object	Requests	Hits	Hits %	Misses	Bytes from misses	Total bytes
/1m	22	18	81.82%	4	2.01 kB	7.38 kB
/users	6	0	0.00%	6	2.6 kB	2.6 kB
/customheader	2	0	0.00%	2	0.96 kB	0.96 kB
/	2	0	0.00%	2	604B	604B
/favicon.ico	1	0	0.00%	0	0B	459B



Filter displayed data: Filter data