

To Enable AWS and Configure a webserver on it using CLI.

OBJECTIVES:

The architecture includes-

- Webserver configured on EC2 Instance
- Document Root(/var/www/html) made persistent by mounting on EBS Block Device.
- Static objects used in code such as pictures stored in S3
- Setting up Content Delivery Network using CloudFront and using the origin domain as S3 bucket.
- Finally place the Cloud Front URL on the webapp code for security and low latency.

Moving towards the solution:

1. Configure AWS on your windows command prompt

First, create an IAM user then proceed.

Command: `aws configure`

```
C:\Users\lenovo\Desktop\keys>aws configure
AWS Access Key ID [*****KLEX]:
AWS Secret Access Key [*****pygB]:
Default region name [us-east-1]:
Default output format [None]:
```

2. Launch the instance

Command: `aws ec2 run-instances --image-id ami-0947d2ba12ee1f75 --instance-type t2.micro --count 1 --subnet-id subnet-eb62cd8d --security-group-ids sg-050f2197a20601d8a --key-name Client`


```
C:\Users\lenovo\Desktop\keys> aws ec2 create-volume --volume-type gp2 --size 1 --availability-zone us-east-1a
{
  "AvailabilityZone": "us-east-1a",
  "CreateTime": "2020-10-29T06:44:28+00:00",
  "Encrypted": false,
  "Size": 1,
  "SnapshotId": "",
  "State": "creating",
  "VolumeId": "vol-0cb1a654680927d3d",
  "Iops": 100,
  "Tags": [],
  "VolumeType": "gp2",
  "MultiAttachEnabled": false
}

C:\Users\lenovo\Desktop\keys>
```

Now the new volume has been created. It is available, but yet not attached. We shall attach it to the instance that we have just launched.

Create Volume Actions											
Filter by tags and attributes or search by keyword											
<input type="checkbox"/>	Name	Volume ID	Size	Volume Type	IOPS	Snapshot	Created	Availability	State	Alarm Status	Attachment Information
<input type="checkbox"/>		vol-0cb1a654...	1 GiB	gp2	100		October 29, 2020 at 12:14:28 PM UTC+5:30	us-east-1a	available	None	
<input type="checkbox"/>		vol-01e3c4ec...	8 GiB	gp2	100	sn...	October 29, 2020 at 12:06:36 PM UTC+5:30	us-east-1a	in-use	None	i-0c6f06e430ccec885...
<input type="checkbox"/>		vol-0818fd8a...	10 GiB	gp2	100	sn...	September 23, 2020 at 10:49:18 PM UTC+5:30	us-east-1a	in-use	None	i-0d72f70953933de2...
<input type="checkbox"/>		vol-0fc80ccc...	10 GiB	gp2	100	sn...	September 23, 2020 at 10:45:48 PM UTC+5:30	us-east-1a	in-use	None	i-0cb26ba3416b7f4b...

4. Attaching volume

Command: `aws ec2 attach-volume --instance-id i-0c6f06e430ccec885 --volume-id vol-0cb1a654680927d3d --device /dev/xvdf` In the second image, volume is in use.

```
C:\Users\lenovo\Desktop\keys> aws ec2 attach-volume --instance-id i-0c6f06e430ccec885 --volume-id vol-0cb1a654680927d3d --device /dev/xvdf
{
  "AttachTime": "2020-10-29T06:49:43.725000+00:00",
  "Device": "/dev/xvdf",
  "InstanceId": "i-0c6f06e430ccec885",
  "State": "attaching",
  "VolumeId": "vol-0cb1a654680927d3d"
}

C:\Users\lenovo\Desktop\keys>
```

Create Volume Actions											
Filter by tags and attributes or search by keyword											
<input type="checkbox"/>	Name	Volume ID	Size	Volume Type	IOPS	Snapshot	Created	Availability	State	Alarm Status	Attachment Information
<input type="checkbox"/>		vol-0cb1a654680927d3d	1 GiB	gp2	100		October 29, 2020 at 12:14:28 PM UTC+5:30	us-east-1a	in-use	None	i-0c6f06e430ccec885...
<input type="checkbox"/>		vol-01e3c4ec3220e13e	8 GiB	gp2	100	sn...	October 29, 2020 at 12:06:36 PM UTC+5:30	us-east-1a	in-use	None	i-0c6f06e430ccec885...
<input type="checkbox"/>		vol-0818fd8a32a3f0e0	10 GiB	gp2	100	sn...	September 23, 2020 at 10:49:18 PM UTC+5:30	us-east-1a	in-use	None	i-0d72f70953933de2...
<input type="checkbox"/>		vol-0fc80ccc36b46166	10 GiB	gp2	100	sn...	September 23, 2020 at 10:45:48 PM UTC+5:30	us-east-1a	in-use	None	i-0cb26ba3416b7f4b...

5. Logging into the instance

Command: `ssh -l ec2-user 54.80.1.209 -i Client.pem sudo fdisk /dev/xvdf`

```
C:\Users\lenovo\Desktop\keys>ssh -l ec2-user 54.80.1.209 -i Client.pem sudo fdisk /dev/xvdf

Welcome to fdisk (util-linux 2.30.2).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0x361aeef.

Command (m for help): _
```

6. Creating partition

```
C:\Users\lenovo\Desktop\keys>ssh -l ec2-user 54.80.1.209 -i Client.pem sudo fdisk /dev/xvdf

Welcome to fdisk (util-linux 2.30.2).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0x361aeef.

Command (m for help): n
Partition type
  p   primary (0 primary, 0 extended, 4 free)
  e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1):
First sector (2048-2097151, default 2048):
Last sector, +sectors or +size{K,M,G,T,P} (2048-2097151, default 2097151): +500M

Created a new partition 1 of type 'Linux' and of size 500 MiB.

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.

C:\Users\lenovo\Desktop\keys>
```

7. Format the partition

Command: `ssh -l ec2-user 54.80.1.209 -i Client.pem sudo mkfs.ext4 /dev/xvdf`

```
C:\Users\lenovo\Desktop\keys>ssh -l ec2-user 54.80.1.209 -i Client.pem sudo mkfs.ext4 /dev/xvdf
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
65536 inodes, 262144 blocks
13107 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=268435456
8 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376

Allocating group tables: done
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done

C:\Users\lenovo\Desktop\keys>
```

8. Mount

To mount in html we need to install apache webserver

Command: `ssh -l ec2-user 54.80.1.209 -i Client.pem sudo yum install httpd -y`

```
Command Prompt
C:\Users\lenovo\Desktop\keys>ssh -l ec2-user 54.80.1.209 -i Client.pem sudo yum install httpd -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
--> Package httpd.x86_64 0:2.4.46-1.amzn2 will be installed
--> Processing Dependency: httpd-tools = 2.4.46-1.amzn2 for package: httpd-2.4.46-1.amzn2.x86_64
--> Processing Dependency: httpd filesystem = 2.4.46-1.amzn2 for package: httpd-2.4.46-1.amzn2.x86_64
--> Processing Dependency: system-logos-httpd for package: httpd-2.4.46-1.amzn2.x86_64
--> Processing Dependency: mod_http2 for package: httpd-2.4.46-1.amzn2.x86_64
--> Processing Dependency: httpd filesystem for package: httpd-2.4.46-1.amzn2.x86_64
--> Processing Dependency: /etc/mime.types for package: httpd-2.4.46-1.amzn2.x86_64
--> Processing Dependency: libaprutil-1.so.0()(64bit) for package: httpd-2.4.46-1.amzn2.x86_64
--> Processing Dependency: libapr-1.so.0()(64bit) for package: httpd-2.4.46-1.amzn2.x86_64
--> Running transaction check
--> Package apr.x86_64 0:1.6.3-5.amzn2.0.2 will be installed
--> Package apr-util.x86_64 0:1.6.1-5.amzn2.0.2 will be installed
--> Processing Dependency: apr-util-bdb(x86_64) = 1.6.1-5.amzn2.0.2 for package: apr-util-1.6.1-5.amzn2.0.2.x86_64
--> Package generic-logos-httpd.noarch 0:18.0.0-4.amzn2 will be installed
--> Package httpd filesystem.noarch 0:2.4.46-1.amzn2 will be installed
--> Package httpd-tools.x86_64 0:2.4.46-1.amzn2 will be installed
--> Package mailcap.noarch 0:2.1.41-2.amzn2 will be installed
--> Package mod_http2.x86_64 0:1.15.14-2.amzn2 will be installed
--> Running transaction check
--> Package apr-util-bdb.x86_64 0:1.6.1-5.amzn2.0.2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

Package Arch Version Repository Size
Installing:
httpd x86_64 2.4.46-1.amzn2 amzn2-core 1.3 M
Installing for dependencies:
apr x86_64 1.6.3-5.amzn2.0.2 amzn2-core 118 k
apr-util x86_64 1.6.1-5.amzn2.0.2 amzn2-core 99 k
apr-util-bdb x86_64 1.6.1-5.amzn2.0.2 amzn2-core 19 k
generic-logos-httpd noarch 18.0.0-4.amzn2 amzn2-core 19 k
httpd filesystem noarch 2.4.46-1.amzn2 amzn2-core 23 k
httpd-tools x86_64 2.4.46-1.amzn2 amzn2-core 87 k
mailcap noarch 2.1.41-2.amzn2 amzn2-core 31 k
mod_http2 x86_64 1.15.14-2.amzn2 amzn2-core 147 k

Transaction Summary
Install 1 Package (+8 Dependent packages)
Total download size: 1.8 M
Installed size: 5.1 M
Downloading packages:
```

To mount the command is: `ssh -l ec2-user 54.80.1.209 -i Client.pem sudo mount /dev/xvdf1 /var/www/html`

```
Complete!
C:\Users\lenovo\Desktop\keys>ssh -l ec2-user 54.80.1.209 -i Client.pem sudo mount /dev/xvdf1 /var/www/html
C:\Users\lenovo\Desktop\keys>
```

Mounted successfully

9. Start webserver and check its status

Command:

```
systemctl start httpd
```

```
systemctl status httpd
```

```
C:\Users\lenovo\Desktop\keys>ssh -l ec2-user 54.80.1.209 -i Client.pem sudo systemctl start httpd
C:\Users\lenovo\Desktop\keys>ssh -l ec2-user 54.80.1.209 -i Client.pem sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
   Active: active (running) since Thu 2020-10-29 07:56:40 UTC; 12s ago
     Docs: man:httpd.service(8)
  Main PID: 665 (httpd)
   Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec:  0 B/sec"
    CGroup: /system.slice/httpd.service
            └─665 /usr/sbin/httpd -DFOREGROUND
              └─666 /usr/sbin/httpd -DFOREGROUND
                └─667 /usr/sbin/httpd -DFOREGROUND
                  └─668 /usr/sbin/httpd -DFOREGROUND
                    └─669 /usr/sbin/httpd -DFOREGROUND
                      └─670 /usr/sbin/httpd -DFOREGROUND

Oct 29 07:56:40 ip-172-31-1-188.ec2.internal systemd[1]: Starting The Apache HTTP Server...
Oct 29 07:56:40 ip-172-31-1-188.ec2.internal systemd[1]: Started The Apache HTTP Server.

C:\Users\lenovo\Desktop\keys>_
```

10. Login to the aws cli

```
C:\Users\lenovo\Desktop\keys>ssh -l ec2-user 54.80.1.209 -i Client.pem
Last login: Thu Oct 29 07:29:58 2020 from 106.198.246.254

  _|_  _|_  )
 _|_ (  _|_ /  Amazon Linux 2 AMI
Last login: Thu Oct 29 07:29:58 2020 from 106.198.246.254

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

https://aws.amazon.com/amazon-linux-2/
25 package(s) needed for security, out of 39 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-1-188 ~]$ _
```

11. Login to the root and then go to html directory

```
[ec2-user@ip-172-31-1-188 ~]$ sudo su - root
Last login: Thu Oct 29 07:30:47 UTC 2020 on pts/0
[root@ip-172-31-1-188 ~]# cd /var/www/html
[root@ip-172-31-1-188 html]#
```

12. Transfer the img and copy it to the directory

```
[root@ip-172-31-1-188 html]# cd /home/ec2-user
[root@ip-172-31-1-188 ec2-user]# ls
sk.jpeg
[root@ip-172-31-1-188 ec2-user]# cp sk.jpeg /var/www/html
[root@ip-172-31-1-188 ec2-user]# cd /var/www/html
[root@ip-172-31-1-188 html]# ls
awscli.html  lost+found  sk.jpeg
[root@ip-172-31-1-188 html]# vi awscli.html
```

13. Now write the html code

```
<html>

<!-- hitwebcounter Code START -->
<a href="https://www.hitwebcounter.com" target="_blank">
</a>
<body bgcolor='aqua'>
<style>
h1 {text-align: center;}
h6 {text-align:center;}
</style>

<h1>Trying aws with cli</h1>

<h6>Working on AWS with cli is far better than the GUI experience!</h6>

<style>
img {
display: block;
margin-left: auto;
margin-right: auto;
}
</style>
<img src='sk.jpeg'>

</html>
```

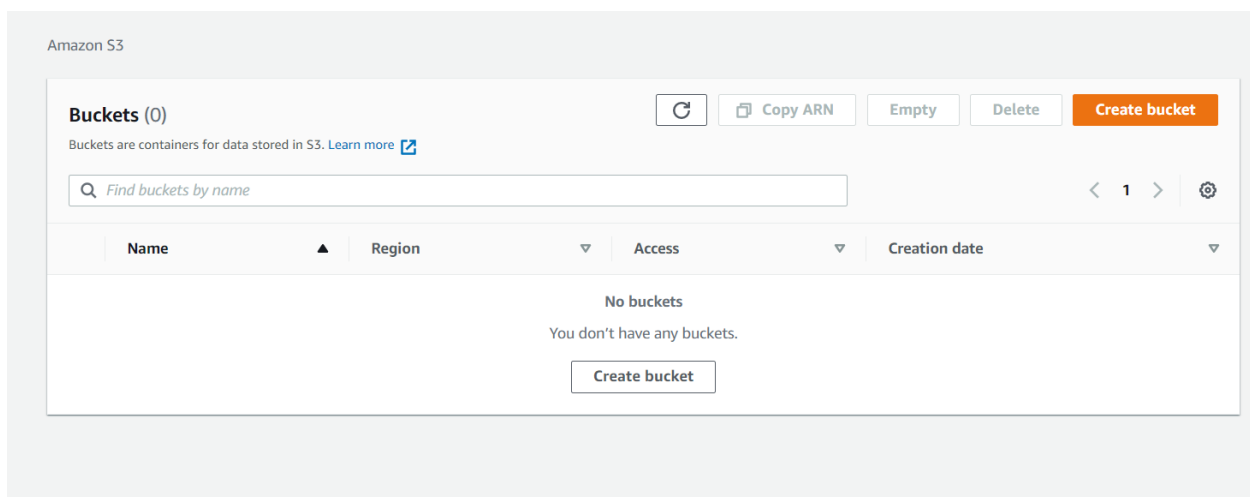
Now we can check the webserver on the browser. On your browser, type your IP and the file name.

However, as EBS is not too reliable, we shall switch to S3 storage. We shall now, put our image in S3 bucket and continue further.

```
[root@ip-172-31-1-188 html]# exit
logout
[ec2-user@ip-172-31-1-188 ~]$
[ec2-user@ip-172-31-1-188 ~]$ exit
logout
Connection to 54.80.1.209 closed.
```

14. come out of instance to make s3 bucket

Currently, there is no bucket.



We shall use the following command to create the bucket.

```
C:\Users\lenovo\Desktop\keys>aws s3api create-bucket --bucket shreya18213 --region us-east-1 --create-bucket-configuration LocationConstraint
{
  "Location": "http://shreya18213.s3.amazonaws.com/"
}
```

Make the bucket public



Command: `aws s3api put-bucket-acl --acl public-read --bucket skcli` (skcli is the bucket name)

```
C:\Users\lenovo\Desktop\keys>aws s3api put-bucket-acl --acl public-read --bucket skcli

C:\Users\lenovo\Desktop\keys>
```

skcli

Bucket overview

Region	Amazon resource name (ARN)	Creation date	Access
US East (N. Virginia) us-east-1	 arn:aws:s3:::skcli	October 29, 2020, 14:37 (UTC+05:30)	 Public

Objects | Properties | **Permissions** | Metrics | Management | Access points

Currently there are no objects as shown below


Objects | Properties | Permissions | Metrics | Management | Access points

Drag and drop files and folders you want to upload here, or choose **Upload**.

Objects (0)

Objects are the fundamental entities stored in Amazon S3. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

    **Upload**

 Find objects by prefix

< 1 > 

	Name	Type	Last modified	Size	Storage class
---	------	------	---------------	------	---------------

No objects

You don't have any objects in this bucket.

Upload

15. Adding objects.


Command: `aws s3api put-object --bucket skcli --key sk.jpeg --body C:\Users\lenovo\Desktop\sk.jpeg`

```
C:\Users\lenovo\Desktop\keys>aws s3api put-object --bucket skcli --key sk.jpeg --body C:\Users\lenovo\Desktop\sk.jpeg
{
  "ETag": "\"430354835778c2d107dd5b30cf85a3a0\""
}
```


Objects (1)

Objects are the fundamental entities stored in Amazon S3. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

    **Upload**

 Find objects by prefix

< 1 > 

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	 sk.jpeg	jpeg	October 29, 2020, 17:24 (UTC+05:30)	85.9 KB	Standard

Object is created. The next step is to make it public

16. Making object public

Command: `aws s3api put-object-acl --bucket skcli --key sk.jpeg --grant-read uri=http://acs.amazonaws.com/groups/global/AllUsers`

```
C:\Users\lenovo\Desktop\keys>aws s3api put-object-acl --bucket skcli --key sk.jpeg --grant-read uri=http://acs.amazonaws.com/groups/global/AllUsers
```

sk.jpeg Latest version ▼

Overview

Properties

Permissions

Select from

Open

Download

Download as

Make public

Copy path

Owner
shreya111kumari

Last modified
Oct 29, 2020 5:39:23 PM GMT+0530

Etag
430354835778c2d107dd5b30cf85a3a0

Storage class
Standard

Server-side encryption
None

Size
85.9 KB

Key
sk.jpeg

Object URL
<https://skcli.s3.amazonaws.com/sk.jpeg>

17. Login to the instance again, and in the HTML file, use the object url

```
<html>

<!-- hitwebcounter Code START -->
<a href="https://www.hitwebcounter.com" target="_blank">
</a>
<body bgcolor='aqua'>
<style>
h1 {text-align: center;}
h6 {text-align:center;}
</style>

<h1>Trying aws with cli</h1>

<h6>Working on AWS with cli is far better than the GUI expe

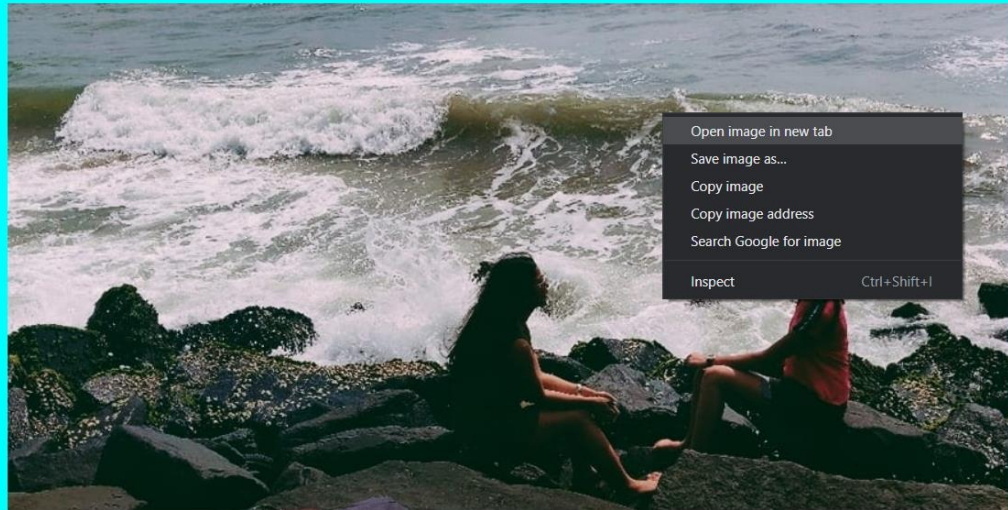
<style>
img {
  display: block;
  margin-left: auto;
  margin-right: auto;
}
</style>
<img src='https://skcli.s3.amazonaws.com/sk.jpeg'>

</html>
-- INSERT --
```

023

Trying aws with cli

Working on AWS with cli is far better than the GUI experience!



18. Using Cloud Front

- To enhance user's experience
- To avoid latency

Command: `aws cloudfront create-distribution --origin-domain-name skcli.s3.amazonaws.com --default-root-object dir-1/sk.jpeg`

```
C:\Users\lenovo\Desktop\keys>aws cloudfront create-distribution --origin-domain-name skcli.s3.amazonaws.com --default-root-object dir-1/sk.jpeg
{
  "Location": "https://cloudfront.amazonaws.com/2019-03-26/distribution/EN0938RL213C5",
  "ETag": "E2FHAWV25MU42",
  "Distribution": {
    "Id": "EN0938RL213C5",
    "ARN": "arn:aws:cloudfront::648825576930:distribution/EN0938RL213C5",
    "Status": "InProgress",
    "LastModifiedTime": "2020-10-29T12:31:31.510000+00:00",
    "InProgressInvalidationBatches": 0,
    "DomainName": "d2u5osxyvwc40g.cloudfront.net",
    "ActiveTrustedSigners": {
      "Enabled": false,
      "Quantity": 0
    }
  }
}
```

Cloud Front distribution is created.

CloudFront Distributions								
<div>Create Distribution Distribution Settings Delete Enable Disable</div>								
Viewing: Any Delivery Method Any State <input type="text"/> << < Viewing 1 to 2 of 2 Items > >>								
Delivery Method	ID	Domain Name	C	Origin	C	Status	State	Last Modified
<input type="checkbox"/> Web	EN0938RL213C5	d2u5osxyvwc40g.cloudfront.net	-	skcli.s3.amazc	-	In Progress	Enabled	2020-10-29 18:01

19. Using domain name of cloudfront for the image:

Paste the domain name of the cloud front in the image source as shown in the image.

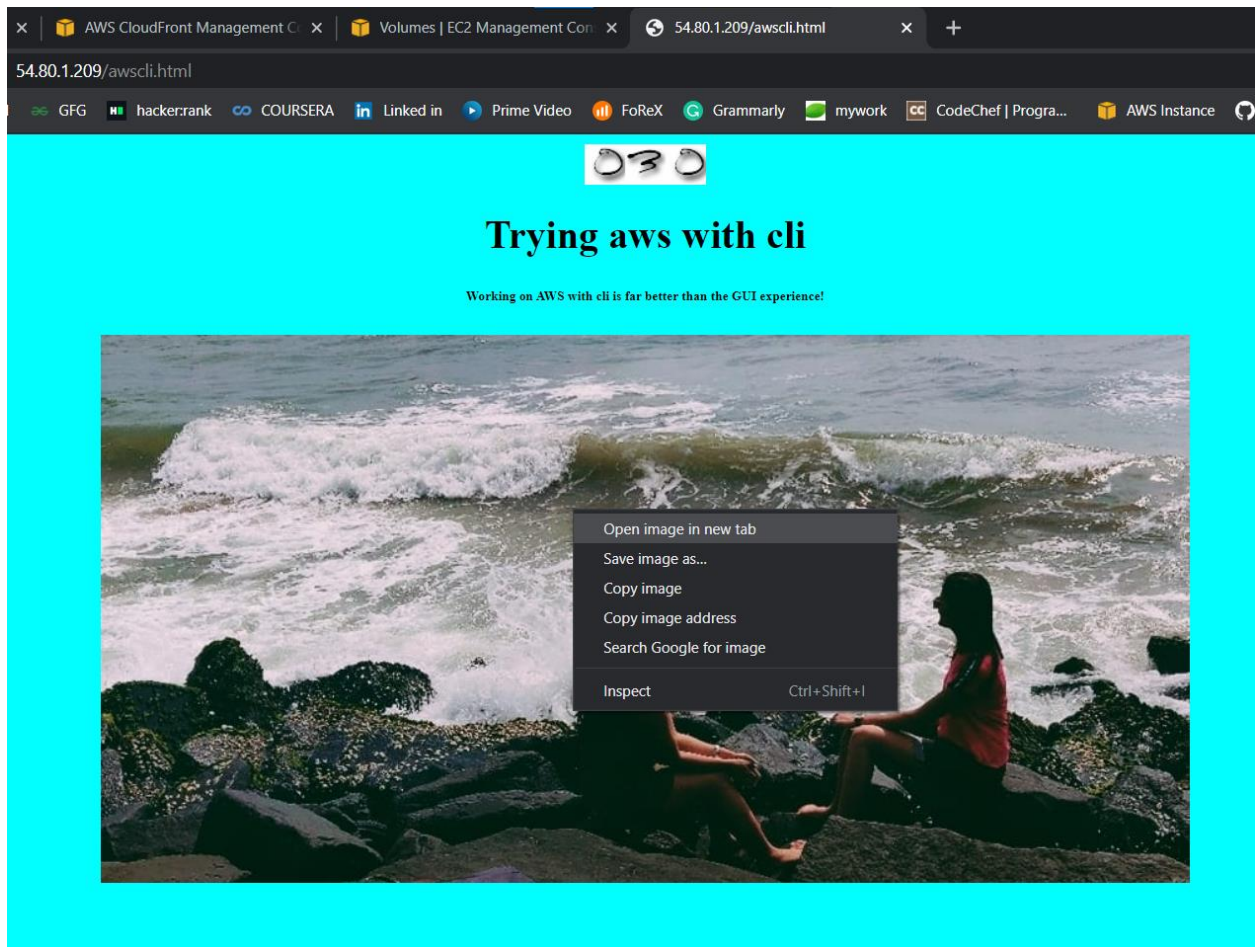
```
<style>
img {
  display: block;
  margin-left: auto;
  margin-right: auto;
}
</style>
<img src='http://d2u5osxyvwc40g.cloudfront.net/sk.jpeg'>

</html>
"awscli.html" 27L, 622C
```

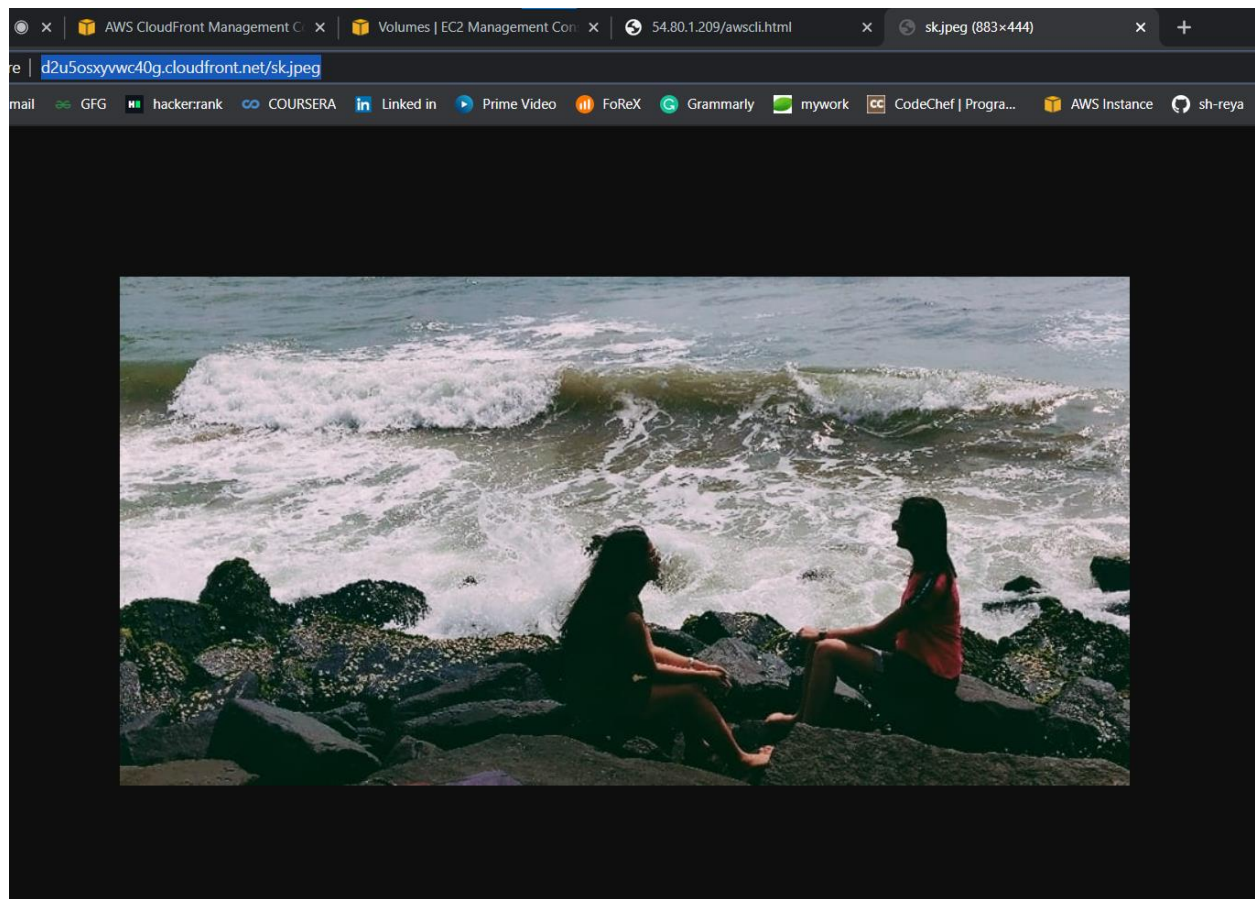
Now, open this file by typing the public ip and the name of the file as shown in the image.

In my case, it is:

54.80.1.209/awscli.html



Open the image in the new tab to see if the link provided is of the cloud front. If yes, the task has been completed successfully!



The url of the image is of the cloudfront. So, here I have completed the task 6 of ARTH successfully.