AWS CSA 2020 LW-World informatic Tasks.

Under the mentorship of Sir Vimal Daga

Task 5

Create High Availability Architecture with AWS CLI

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.

Solution

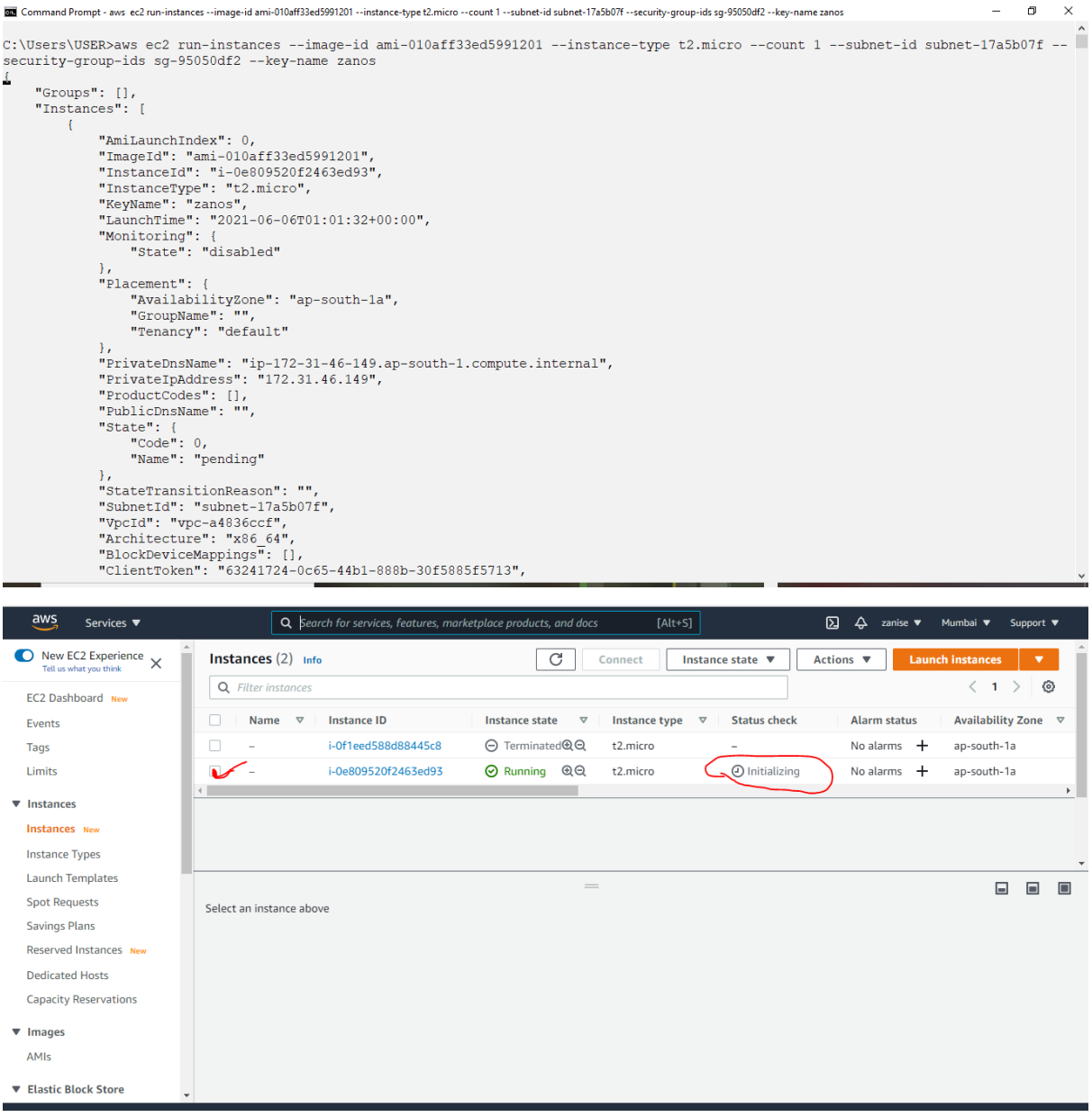
Create High Availability Architecture with AWS CLI

The architecture includes -

- Webserver configured on EC2 Instance

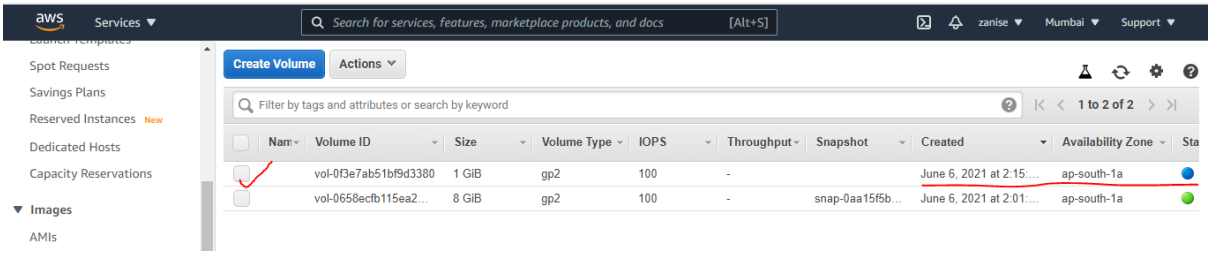
- Document Root(/var/www/html) made persistent by mounting on EBS Block Device.

1. lets create an ec2 instance



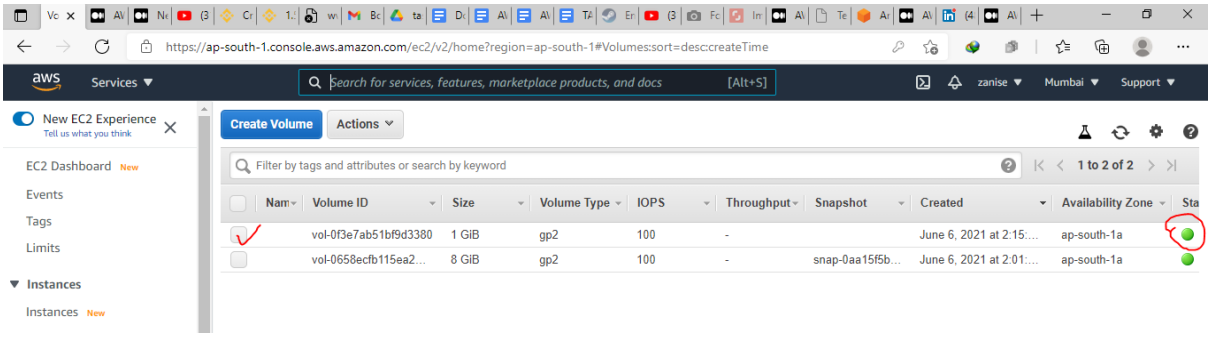
2. lets create a volume (EBS)

```
Command Prompt

C:\Users\USER>aws ec2 create-volume --volume-type gp2 --size 1 --availability-zone ap-south-1a
{
    "AvailabilityZone": "ap-south-1a",
    "CreateTime": "2021-06-06T01:15:26+00:00",
    "Encrypted": false,
    "Size": 1,
    "SnapshotId": "",
    "State": "creating",
    "VolumeId": "vol-0f3e7ab51bf9d3380",
    "Iops": 100,
    "Tags": [],
    "VolumeType": "gp2",
    "MultiAttachEnabled": false
}


C:\Users\USER>
```



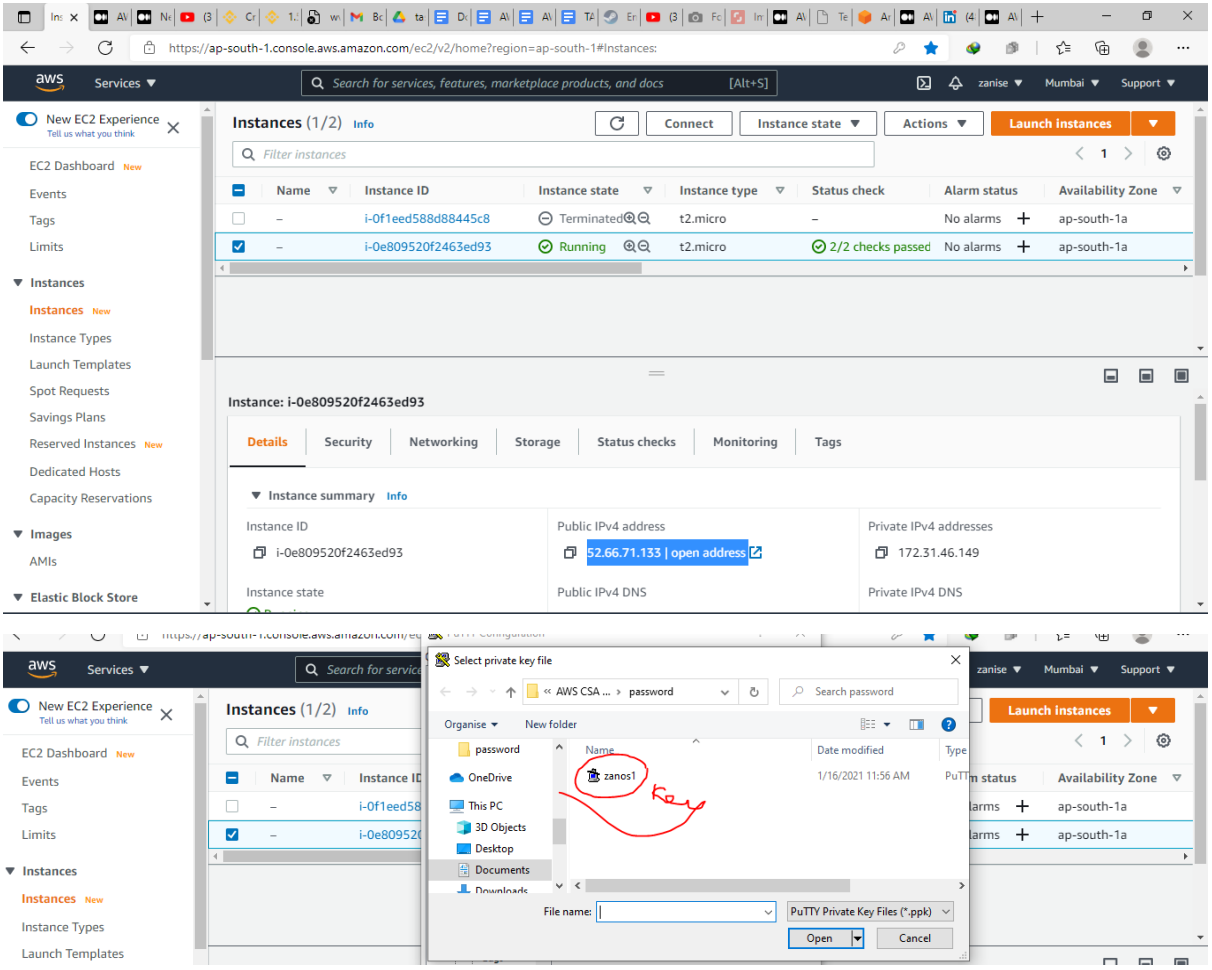
3. Attaching EBS to EC2

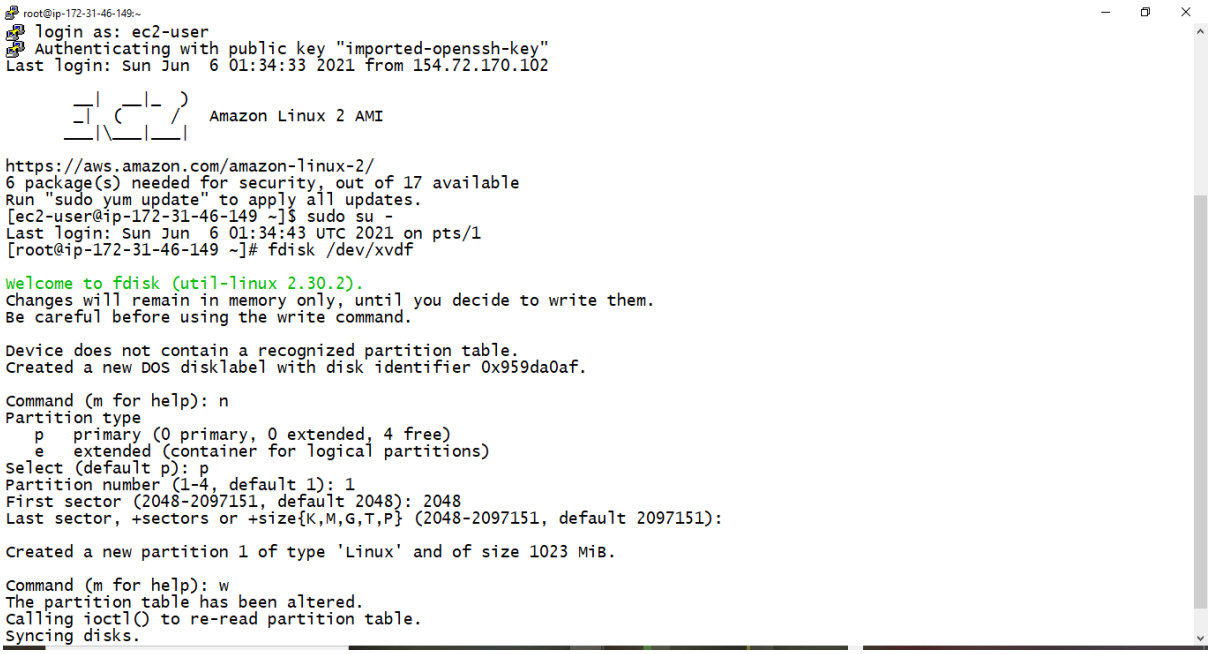
```
Command Prompt

C:\Users\USER>aws ec2 attach-volume --instance-id i-0e809520f2463ed93 --device /dev/sdf --volume-id vol-0f3e7ab51bf9d3380
{
    "AttachTime": "2021-06-06T01:22:31.969000+00:00",
    "Device": "/dev/sdf",
    "InstanceId": "i-0e809520f2463ed93",
    "State": "attaching",
    "VolumeId": "vol-0f3e7ab51bf9d3380"
}


C:\Users\USER>
```



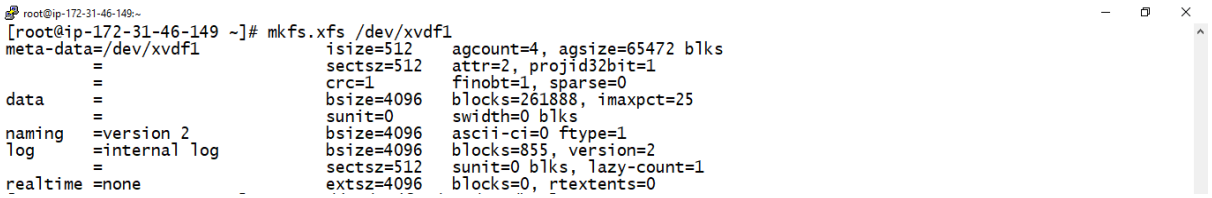
4- Formating EBS partition for that we will need to sshinto the instance using putty. for that we need the instane

- Public IPv4
- private key

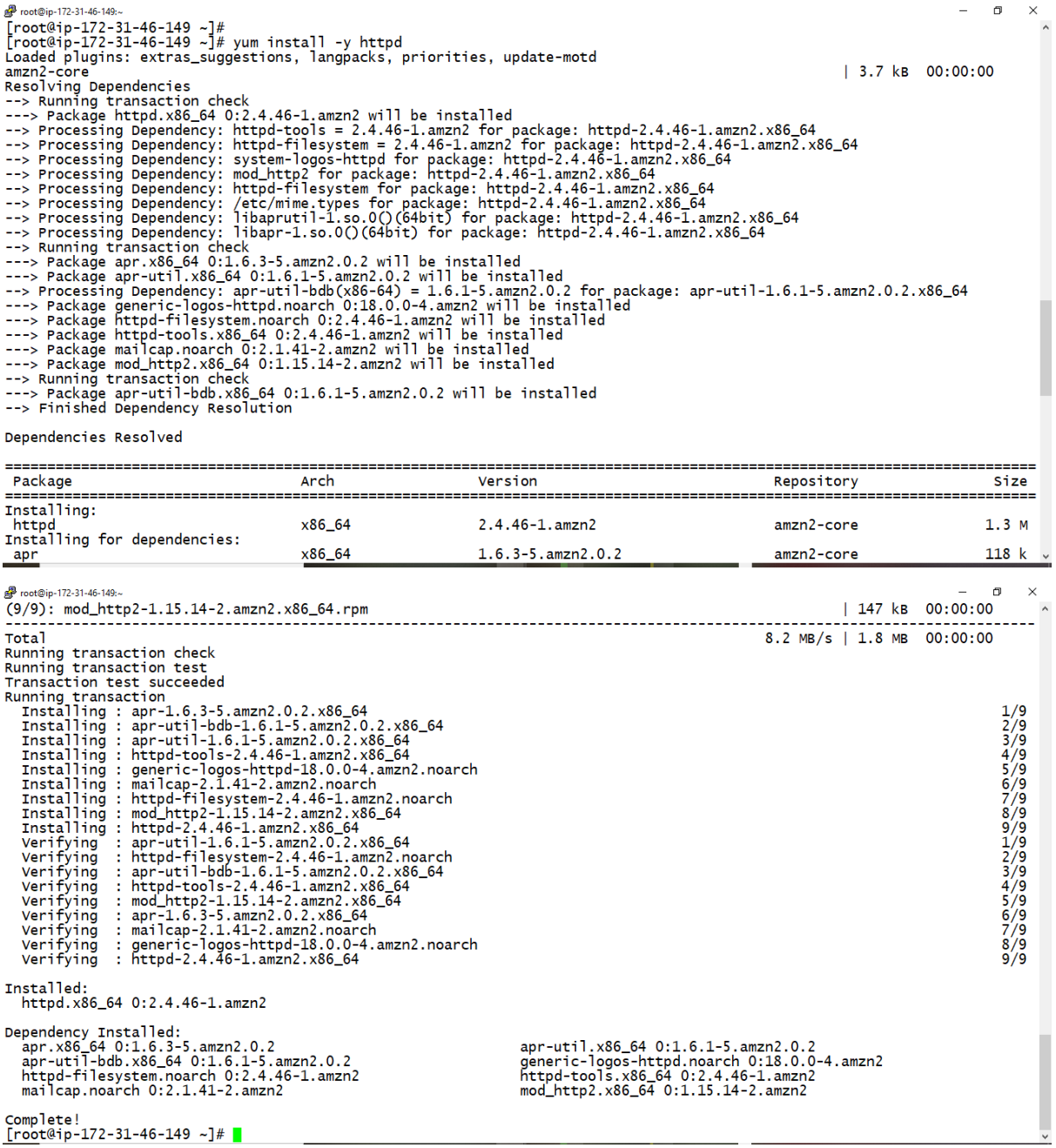




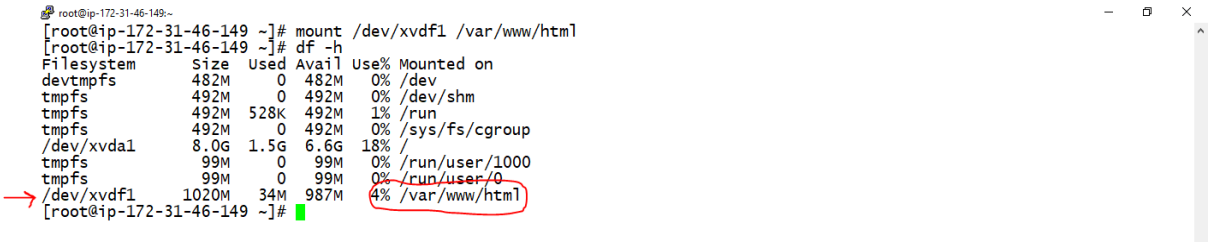
5. lets format the partition created



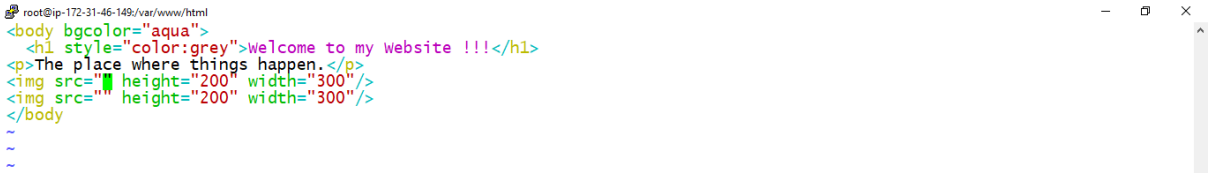
6. lets install apache webserver



7. lets mount /dev/xvdf1 on /var/www/html

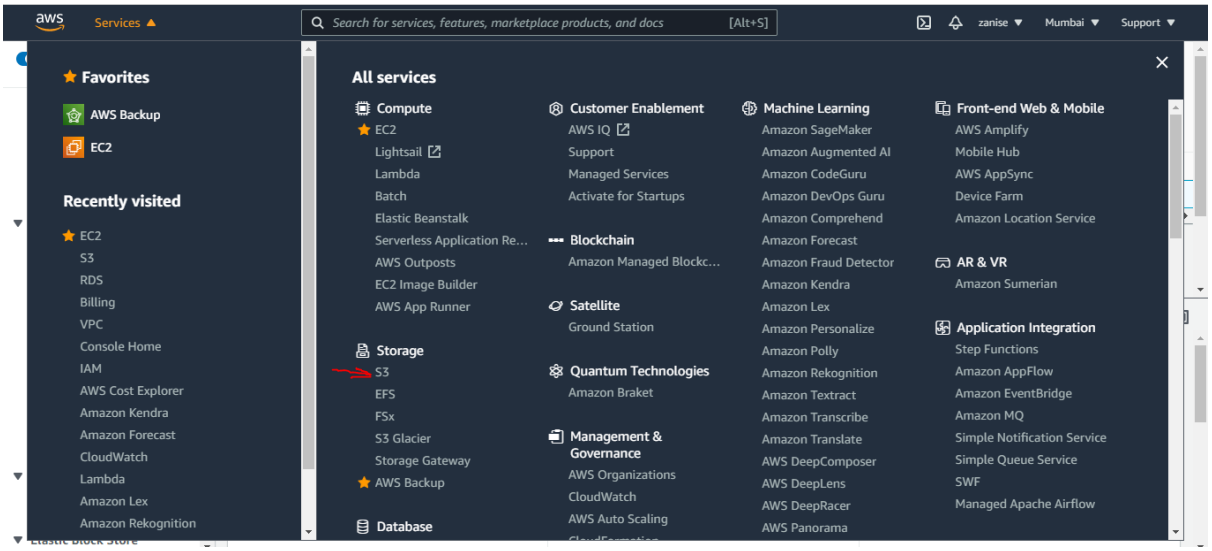


8. Writing a webapp code in the /var/www/html folder

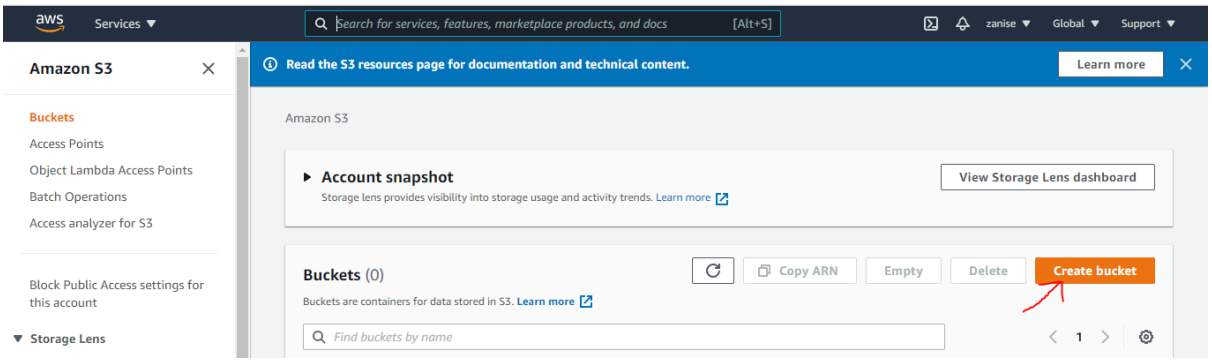


9- Lets create an s3 bucket then upload pictures there

go to service then click on s3

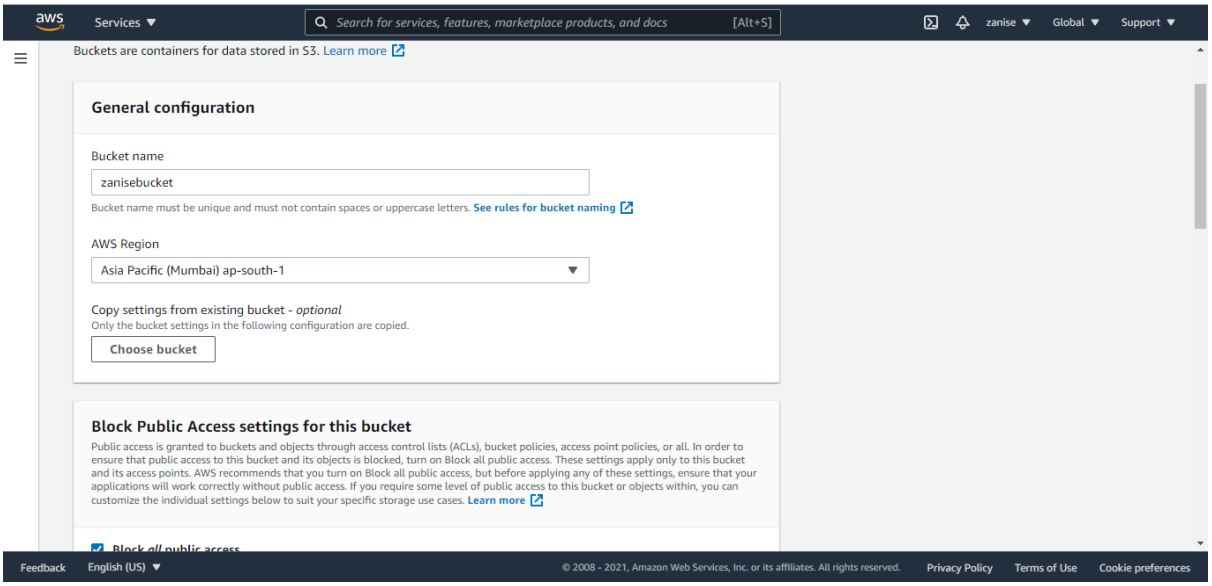


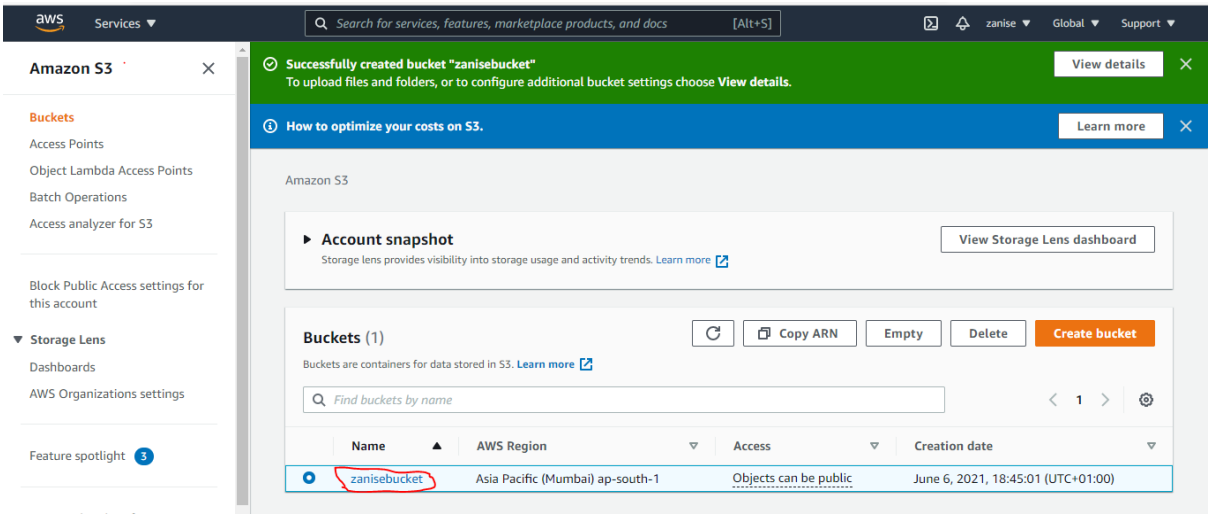
click on create bucket



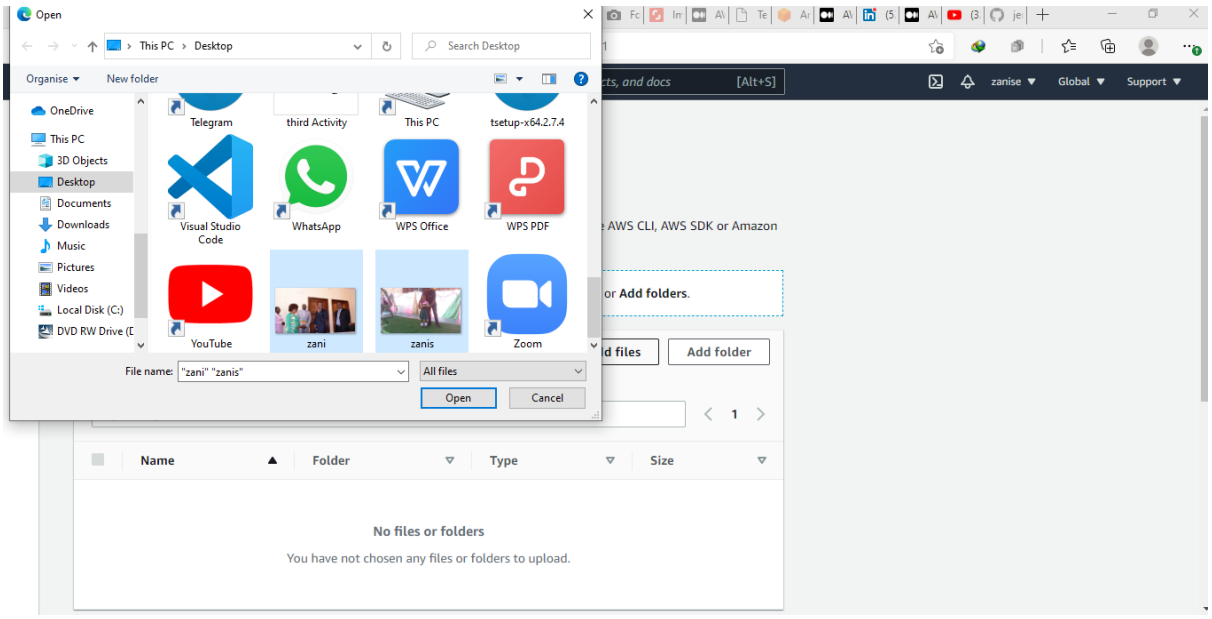
fill the bucket name and the aws region that is all

then click on create bucket

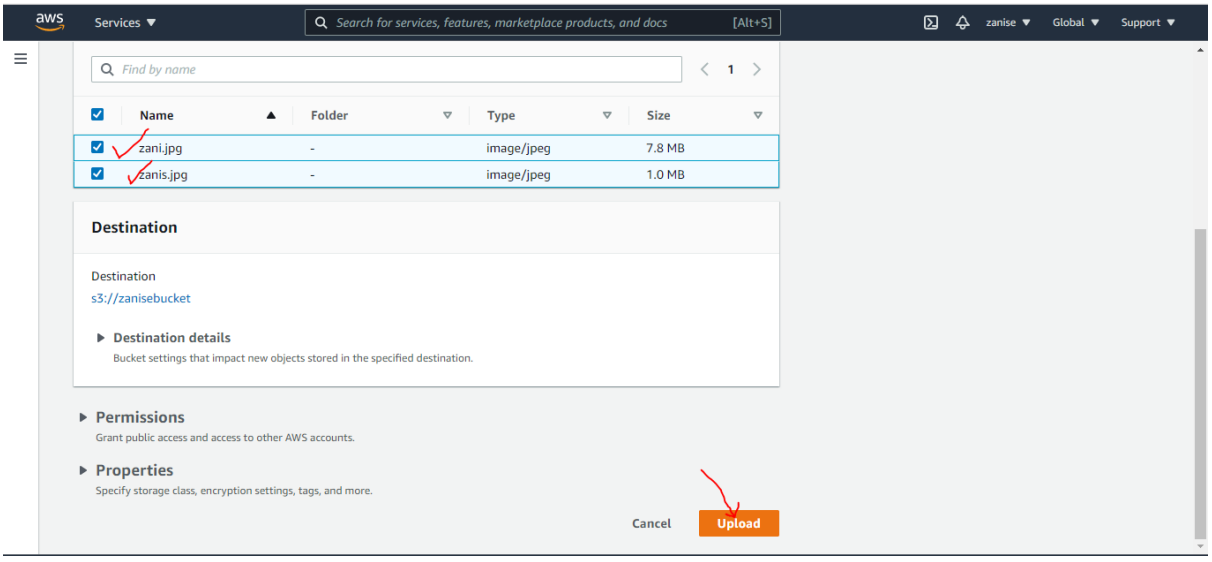


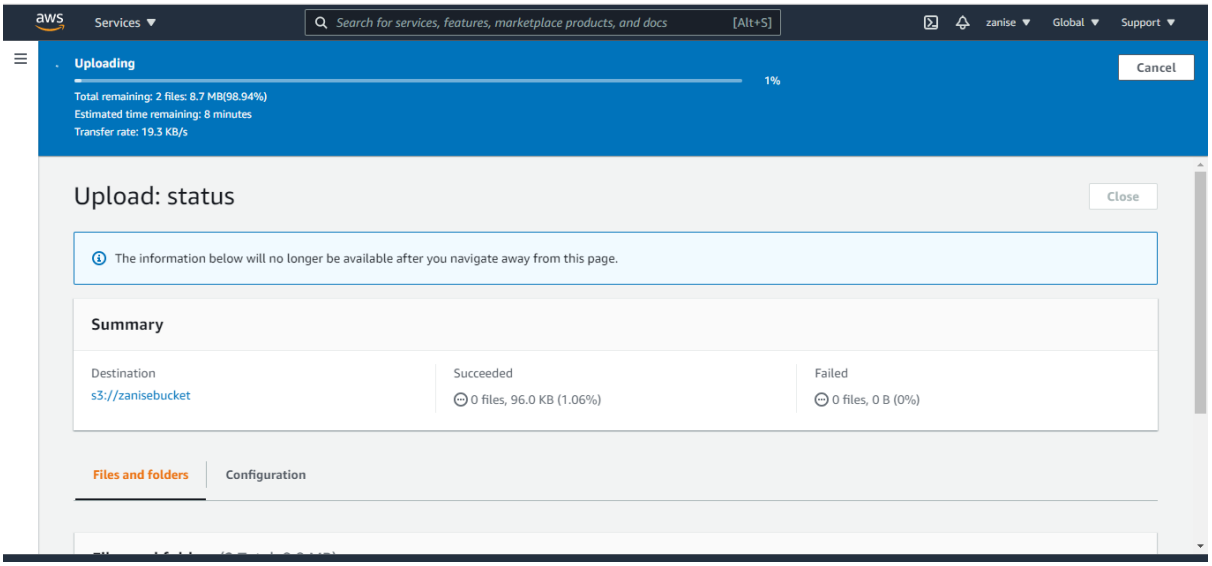


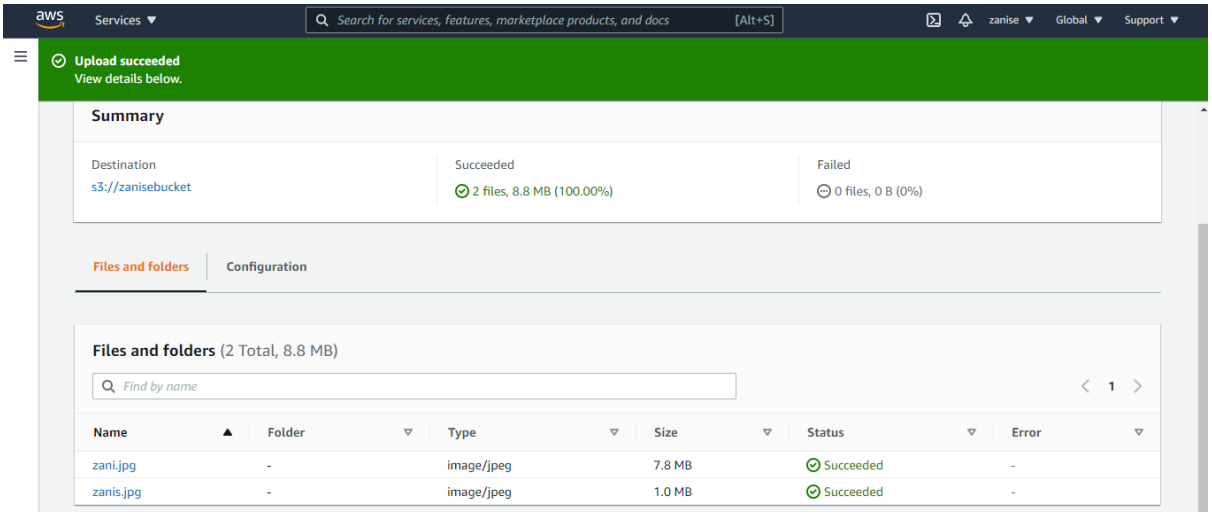
Lets upload photos to the bucket



After choosing the pics to upload click on upload



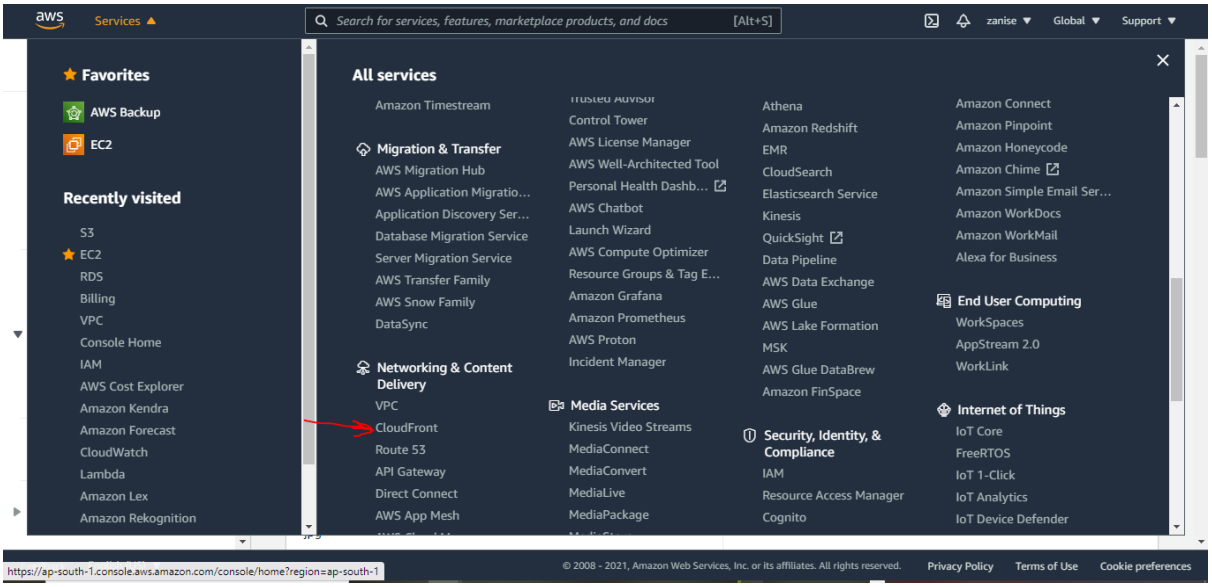




https://zanisebucket.s3.ap-south-1.amazonaws.com/zani.jpg this the url for zani.jpg

https://zanisebucket.s3.ap-south-1.amazonaws.com/zanis.jpg this the url for zanis.jpg

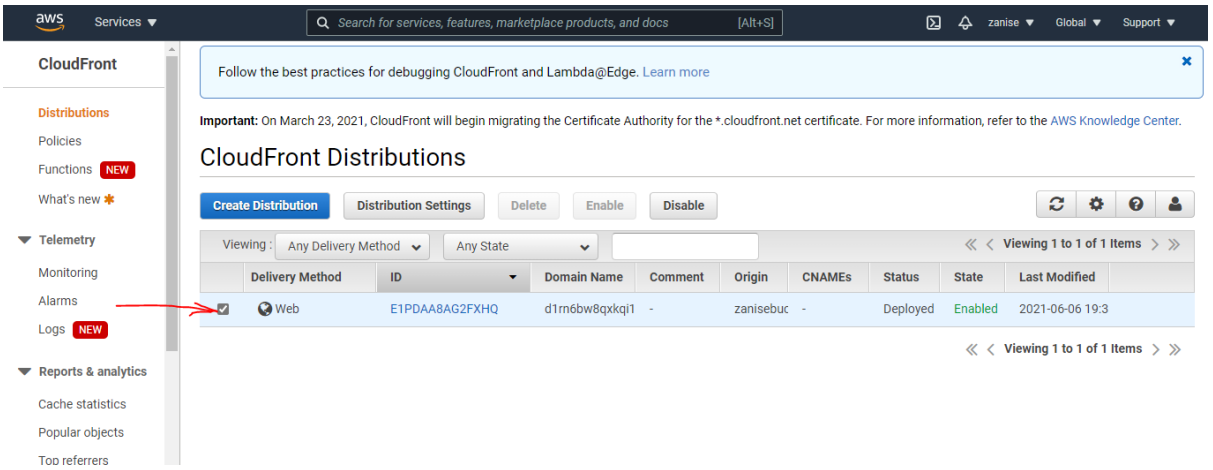
10. Setting up Content Delivery Network using CloudFront and using the origin domain as S3 bucket

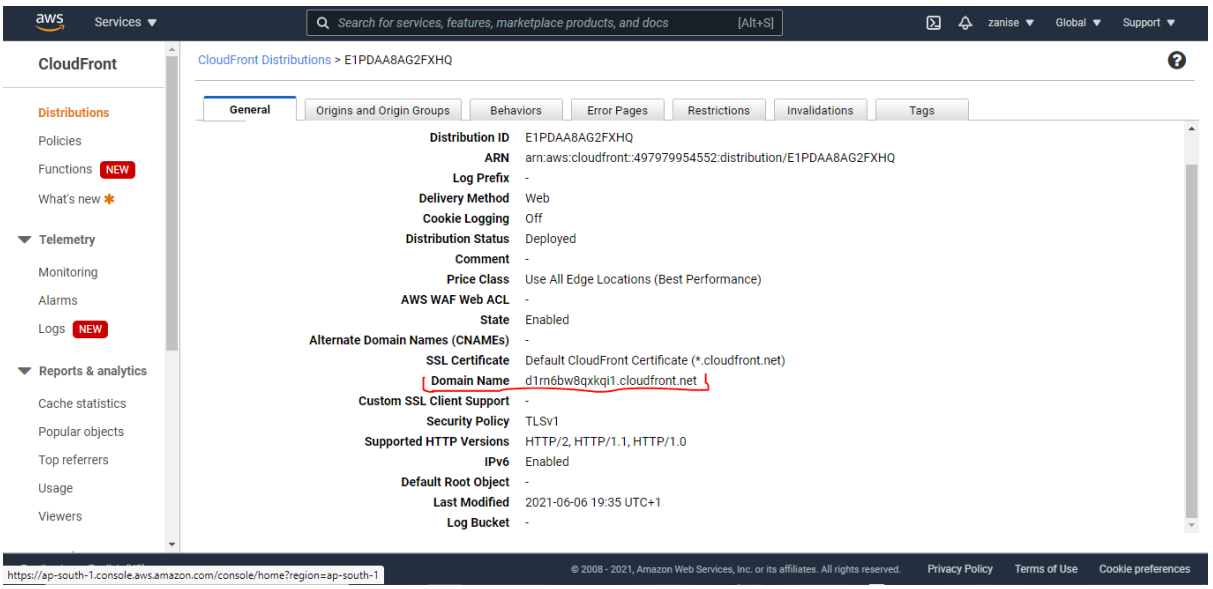


lets create a distribution

Command Prompt - aws cloudfront create-distribution --origin-domain-name zanisebucket.s3.amazonaws.com

```
C:\Users\USER>aws cloudfront create-distribution --origin-domain-name zanisebucket.s3.amazonaws.com
{
    "Location": "https://cloudfront.amazonaws.com/2019-03-26/distribution/E1PDAA8AG2FXHQ",
    "ETag": "E1WFB7ZSVKCBGC",
    "Distribution": {
        "Id": "E1PDAA8AG2FXHQ",
        "ARN": "arn:aws:cloudfront::497979954552:distribution/E1PDAA8AG2FXHQ",
        "Status": "InProgress",
        "LastModifiedTime": "2021-06-06T18:35:19.932000+00:00",
        "InProgressInvalidationBatches": 0,
        "DomainName": "d1rn6bw8qxkqi1.cloudfront.net",
        "ActiveTrustedSigners": {
            "Enabled": false,
            "Quantity": 0
        },
        "DistributionConfig": {
            "CallerReference": "cli-1623004517-972839",
            "Aliases": {
                "Quantity": 0
            },
            "DefaultRootObject": "",
            "Origins": {
                "Quantity": 1,
                "Items": [
                    {
                        "Id": "zanisebucket.s3.amazonaws.com-1623004517-221776",
                        "DomainName": "zanisebucket.s3.amazonaws.com",
                        "OriginPath": "",
                        "CustomHeaders": {
                            "Quantity": 0
                        },
                        "S3OriginConfig": {
                            "OriginAccessIdentity": ""
                        }
```



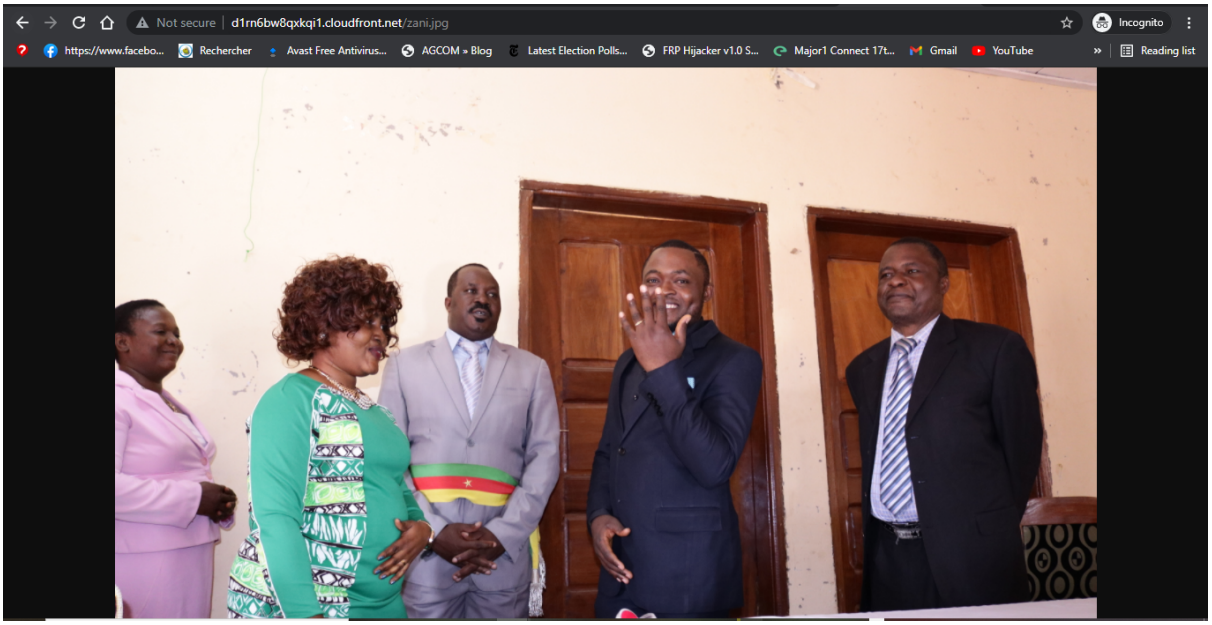


d1rn6bw8qxkqi1.cloudfront.net/zani.jpg

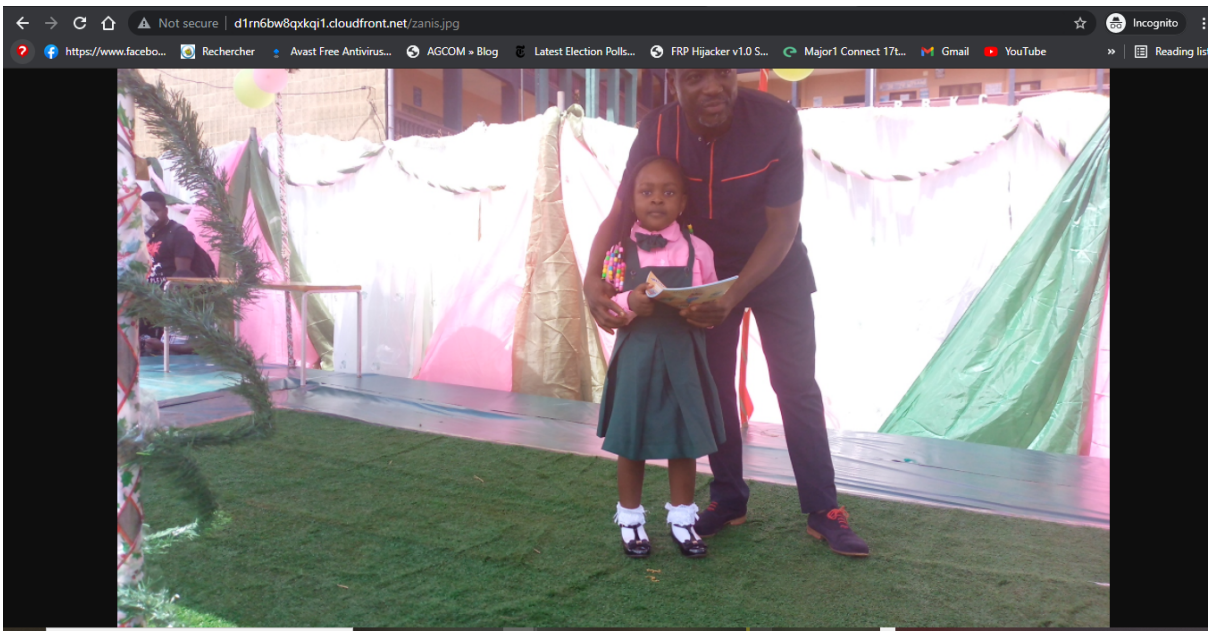
d1rn6bw8qxkqi1.cloudfront.net/zanis.jpg

lets test and see if it works

1. https://d1rn6bw8qxkqi1.cloudfront.net/zani.jpg



2. https://d1rn6bw8qxkqi1.cloudfront.net/zanis.jpg



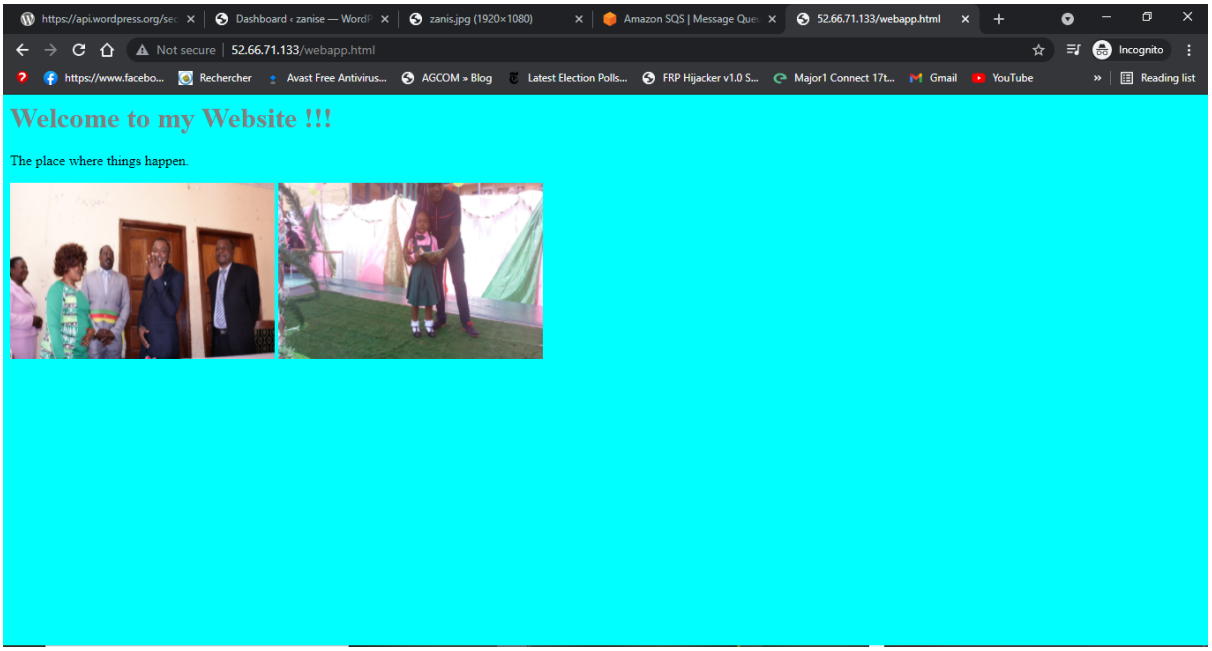
11. So lets Finally place the Cloud Front URL on the webapp code for security and low latency.

```
root@ip-172-31-46-149:/var/www/html
<body bgcolor="aqua">
  <h1 style="color:grey">Welcome to my Website !!!</h1>
<p>The place where things happen.</p>
<img src="d1rn6bw8qxkqi1.cloudfront.net/zanis.jpg" height="200" width="300"/>
<img src="d1rn6bw8qxkqi1.cloudfront.net/zani.jpg" height="200" width="300"/>
</body
~
~
~
~
~
~
```

Lets test to see if it works

using the instance IP and the webapp.html

http://http://52.66.71.133/webapp.html before this you have to edit the inbounds rule and restart the apache webserver



It work perfectly thanks to Sir Vimal and the linux world informatics team

#worldrecordholder #makingindiafutureready #cloudcomputing #aws#awscloud #awscsa2020 #rightmentor #deepknowledge #linuxworld #vimaldaga #righteducation

zanise khan

Am a young passionate of IT

More from zanise khan

Am a young passionate of IT

About

Help

Legal