

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

```

Command Prompt - aws ec2 run-instances --image-id ami-010aff33ed5991201 --instance-type t2.micro --count 1 --subnet-id subnet-17a5b07f --security-group-ids sg-95050df2 --key-name zanos
C:\Users\USER>aws ec2 run-instances --image-id ami-010aff33ed5991201 --instance-type t2.micro --count 1 --subnet-id subnet-17a5b07f --security-group-ids sg-95050df2 --key-name zanos
{
    "Groups": [],
    "Instances": [
        {
            "AmilaunchIndex": 0,
            "ImageId": "ami-010aff33ed5991201",
            "InstanceId": "i-0e809520f2463ed93",
            "InstanceType": "t2.micro",
            "KeyName": "zanos",
            "LaunchTime": "2021-06-06T01:01:32+00:00",
            "Monitoring": {
                "State": "disabled"
            },
            "Placement": {
                "AvailabilityZone": "ap-south-1a",
                "GroupName": "",
                "Tenancy": "default"
            },
            "PrivateDnsName": "ip-172-31-46-149.ap-south-1.compute.internal",
            "PrivateIpAddress": "172.31.46.149",
            "ProductCodes": [],
            "PublicDnsName": "",
            "State": {
                "Code": 0,
                "Name": "pending"
            },
            "StateTransitionReason": "",
            "SubnetId": "subnet-17a5b07f",
            "VpcId": "vpc-a4836ccf",
            "Architecture": "x86_64",
            "BlockDeviceMappings": [],
            "ClientToken": "63241724-0c65-44b1-880b-30f5885f5713",
            "RootDeviceType": "Amazon EBS"
        }
    ]
}

```

The screenshot shows the AWS EC2 Instances page. The left sidebar has sections for EC2 Dashboard, Events, Tags, Limits, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, and Elastic Block Store. The main area displays a table titled 'Instances (2)'. The first instance is 'Terminated' (highlighted with a red circle). The second instance is 'Running' (highlighted with a red circle) and its status is 'Initializing'. The table columns include Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability Zone.

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input checked="" type="checkbox"/>	-	i-0f1eed588d88445c8	Terminated	t2.micro	-	No alarms	ap-south-1a
<input checked="" type="checkbox"/>	-	i-0e809520f2463ed93	Running	t2.micro	Initializing	No alarms	ap-south-1a

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>

```

The screenshot shows the AWS Cloud9 environment with the command prompt window open. The command `aws ec2 create-volume` has been run, and the output shows a new volume being created with ID `vol-0f3e7ab51bf9d3380`. The volume is 1 GiB in size, type gp2, and IOPS 100. It is in the `creating` state.

Name	Volume ID	Size	Type	IOPS	Throughput	Snapshot	Created	Availability Zone	Status
vol-0f3e7ab51bf9d3380	1 GiB	gp2	100	-	-	snap-0aa15f5b...	June 6, 2021 at 2:15...	ap-south-1a	Creating
vol-0658ecfb115ea2...	8 GiB	gp2	100	-	-	snap-0aa15f5b...	June 6, 2021 at 2:01...	ap-south-1a	Available

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>

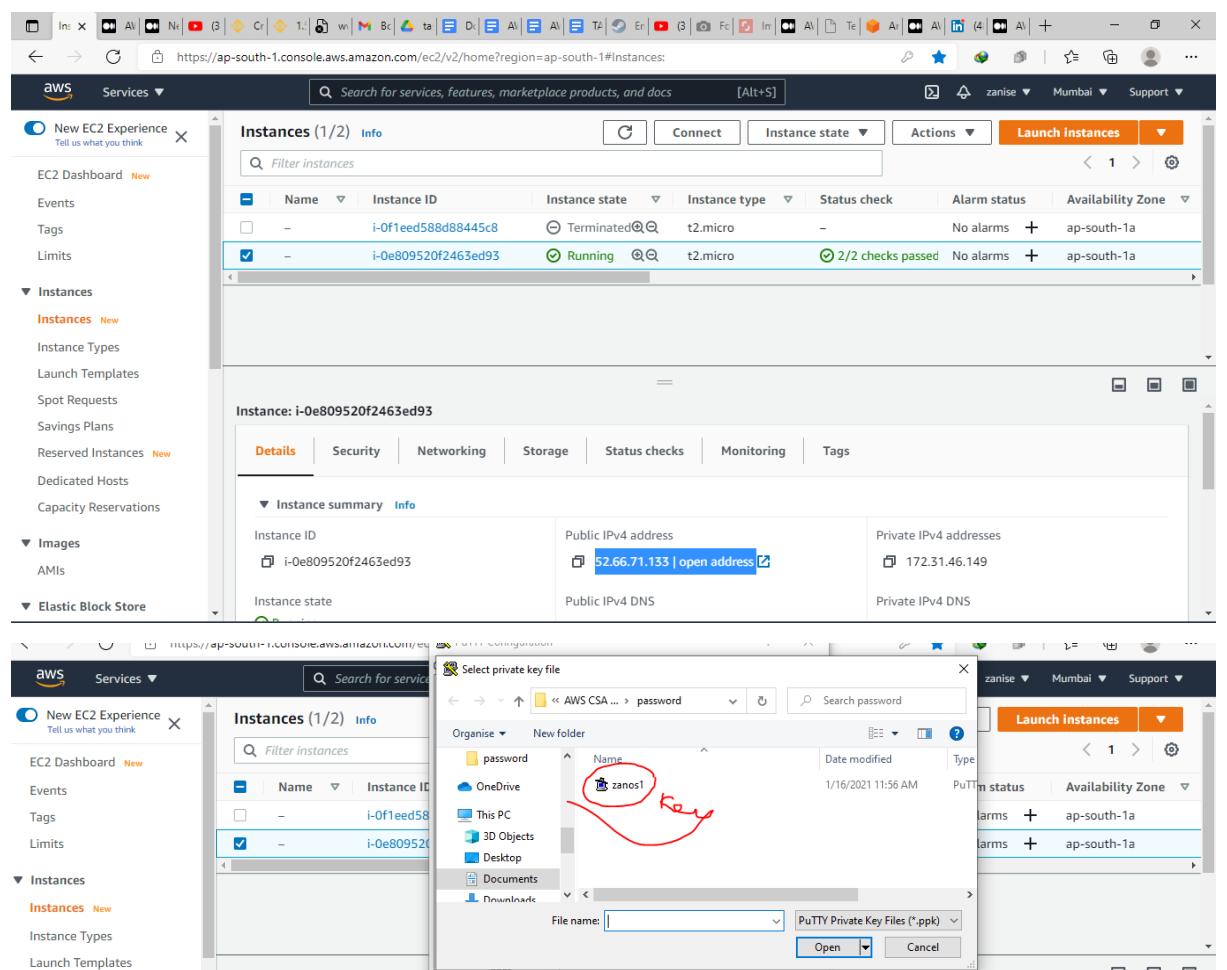
```

The screenshot shows the AWS Cloud9 environment with the command prompt window open. The command `aws ec2 attach-volume` has been run, attaching the volume `vol-0f3e7ab51bf9d3380` to the EC2 instance `i-0e809520f2463ed93` at the device `/dev/sdf`. The volume is now in the `attaching` state.

Name	Volume ID	Size	Type	IOPS	Throughput	Snapshot	Created	Availability Zone	Status
vol-0f3e7ab51bf9d3380	1 GiB	gp2	100	-	-	snap-0aa15f5b...	June 6, 2021 at 2:15...	ap-south-1a	Attaching
vol-0658ecfb115ea2...	8 GiB	gp2	100	-	-	snap-0aa15f5b...	June 6, 2021 at 2:01...	ap-south-1a	Available

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



```
root@ip-172-31-46-149:~#
[ec2-user] Login as: ec2-user
[ec2-user] Authenticating with public key "imported-openssh-key"
Last login: Sun Jun  6 01:34:33 2021 from 154.72.170.102
              _\|_/_ 
              _\|_|_ Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
6 package(s) needed for security, out of 17 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-46-149 ~]$ sudo su -
Last login: Sun Jun  6 01:34:43 UTC 2021 on pts/1
[root@ip-172-31-46-149 ~]# 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 0x959da0af.

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): 1
First sector (2048-2097151, default 2048): 2048
Last sector, +sectors or +size{K,M,G,T,P} (2048-2097151, default 2097151):
Created a new partition 1 of type 'Linux' and of size 1023 MiB.

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

5. lets format the partition created

```
[root@ip-172-31-46-149 ~]# mkfs.xfs /dev/xvdf1
meta-data=/dev/xvdf1      isize=512    agcount=4, agsize=65472 blks
                          =                      sectsz=512  attr=2, projid32bit=1
                          =                      crc=1    finobt=1, sparse=0
data        =            bsize=4096   blocks=261888, imaxpct=25
                          =                      sunit=0    swidth=0 blks
naming      =version_2    bsize=4096   ascii-ci=fftype=1
log         =internal log  bsize=4096   blocks=855, version=2
                          =                      sectsz=512  sunit=0 blks, lazy-count=1
realtime    =none          extsz=4096   blocks=0, rtextents=0
```

6. lets install apache webserver

```

root@ip-172-31-46-149:~# [root@ip-172-31-46-149 ~]# yum install -y httpd
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
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: libaprutil1.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

(9/9): mod_http2-1.15.14-2.amzn2.x86_64.rpm
-----
Total                                         8.2 MB/s | 1.8 MB 00:00:00
-----[root@ip-172-31-46-149:~]# [root@ip-172-31-46-149 ~]# mod_http2-1.15.14-2.amzn2.x86_64.rpm
Total
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : apr-1.6.3-5.amzn2.0.2.x86_64          1/9
  Installing : apr-util-bdb-1.6.1-5.amzn2.0.2.x86_64  2/9
  Installing : apr-util-1.6.1-5.amzn2.0.2.x86_64     3/9
  Installing : httpd-tools-2.4.46-1.amzn2.x86_64     4/9
  Installing : generic-logos-httpd-18.0.0-4.amzn2.noarch 5/9
  Installing : mailcap-2.1.41-2.amzn2.noarch         6/9
  Installing : httpd-filesystem-2.4.46-1.amzn2.noarch 7/9
  Installing : mod_http2-1.15.14-2.amzn2.x86_64       8/9
  Installing : httpd-2.4.46-1.amzn2.x86_64          9/9
  Verifying   : apr-1.6.3-5.amzn2.0.2.x86_64          1/9
  Verifying   : httpd-filesystem-2.4.46-1.amzn2.noarch 2/9
  Verifying   : apr-util-bdb-1.6.1-5.amzn2.0.2.x86_64  3/9
  Verifying   : httpd-tools-2.4.46-1.amzn2.x86_64     4/9
  Verifying   : mod_http2-1.15.14-2.amzn2.x86_64       5/9
  Verifying   : apr-1.6.3-5.amzn2.0.2.x86_64          6/9
  Verifying   : mailcap-2.1.41-2.amzn2.noarch         7/9
  Verifying   : generic-logos-httpd-18.0.0-4.amzn2.noarch 8/9
  Verifying   : httpd-2.4.46-1.amzn2.x86_64          9/9

Installed:
httpd.x86_64 0:2.4.46-1.amzn2

Dependency Installed:
apr.x86_64 0:1.6.3-5.amzn2.0.2
apr-util-bdb.x86_64 0:1.6.1-5.amzn2.0.2
httpd-filesystem.noarch 0:2.4.46-1.amzn2
mailcap.noarch 0:2.1.41-2.amzn2

Complete!
[root@ip-172-31-46-149 ~]#

```

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

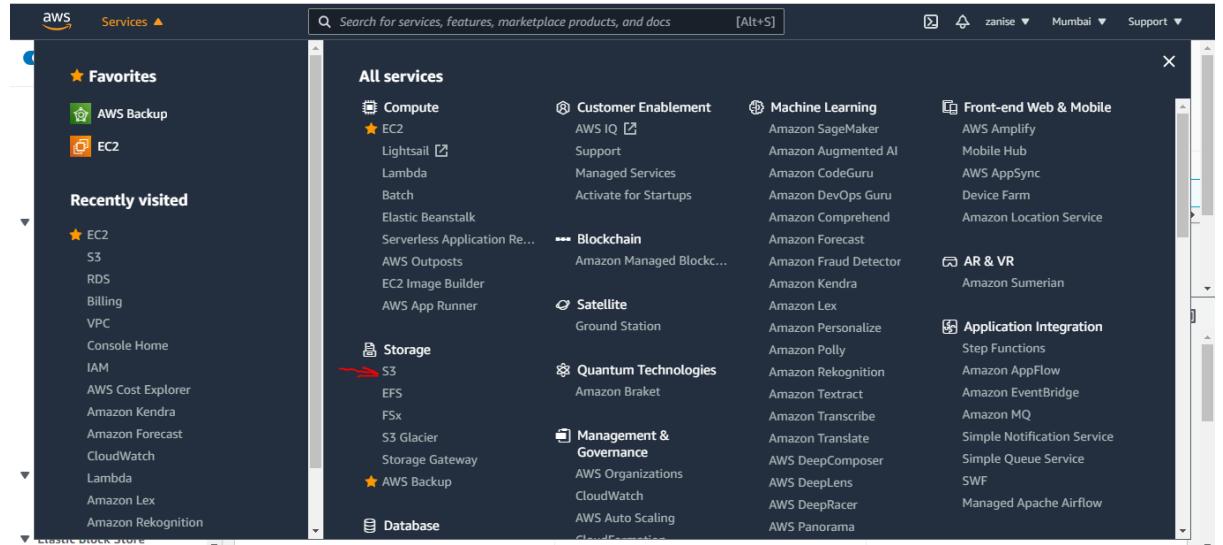
```
root@ip-172-31-46-149:~# mount /dev/xvdf1 /var/www/html
root@ip-172-31-46-149:~# df -h
Filesystem      Size   Used  Avail Use% Mounted on
/devtmpfs        482M    0  482M  0% /dev
tmpfs           492M    0  492M  0% /dev/shm
tmpfs           492M  528K  492M  1% /run
tmpfs           492M    0  492M  0% /sys/fs/cgroup
/dev/xvda1       8.0G  1.5G  6.6G 18% /
tmpfs           99M    0   99M  0% /run/user/1000
tmpfs           99M    0   99M  0% /run/user/0
→ /dev/xvdf1    1020M   34M  987M  4% /var/www/html
[root@ip-172-31-46-149:~]#
```

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

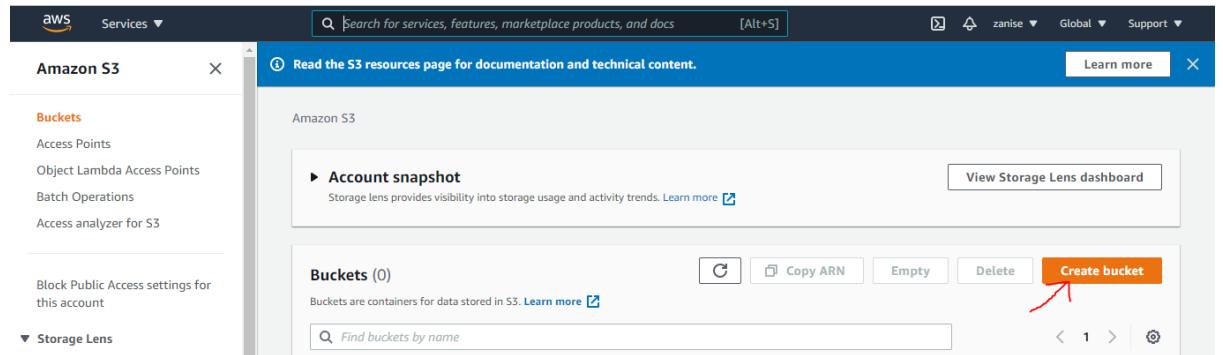
```
root@ip-172-31-46-149:/var/www/html
<body bgcolor="aqua">
  <h1 style="color:grey">welcome to my website !!!</h1>
  <p>The place where things happen.</p>
  
  
</body>
~
```

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

The screenshot shows two consecutive screenshots of the AWS S3 console.

Screenshot 1: General configuration

- Bucket name: zanisebucket
- AWS Region: Asia Pacific (Mumbai) ap-south-1
- Copy settings from existing bucket - optional: Choose bucket

Screenshot 2: Block Public Access settings for this bucket

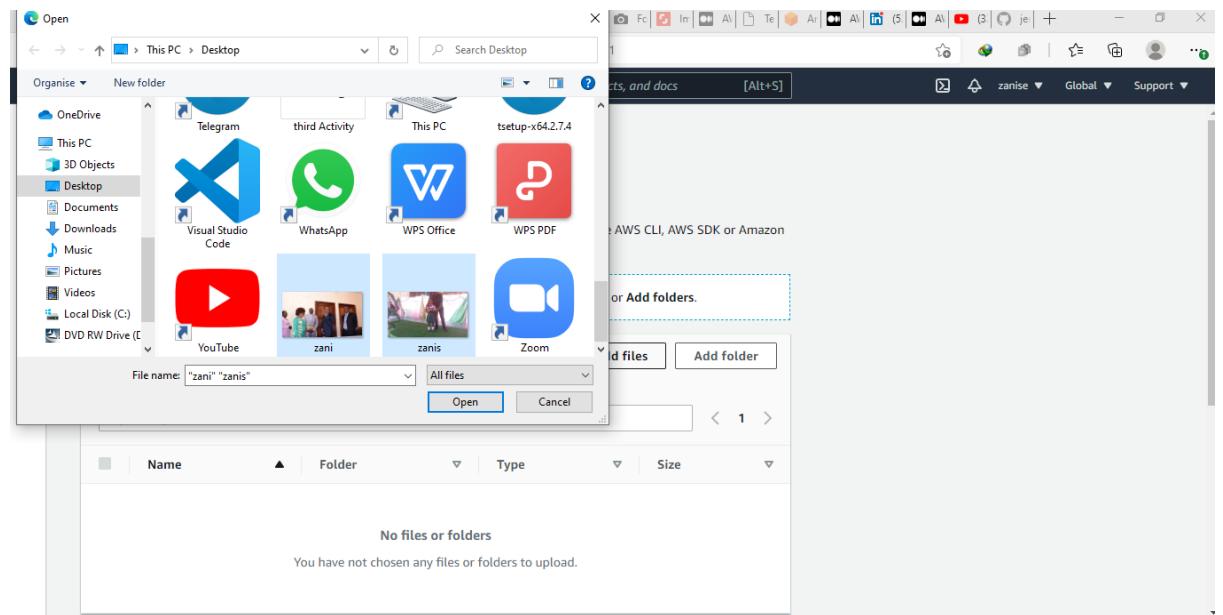
- Block all public access: checked

Success message: Successfully created bucket "zanisebucket". To upload files and folders, or to configure additional bucket settings choose View details.

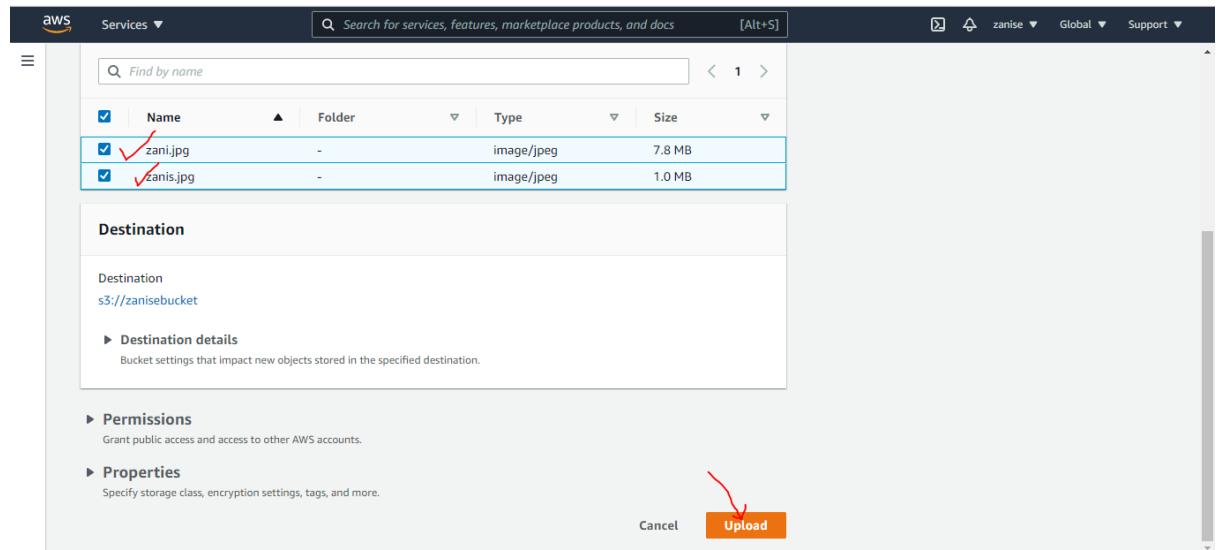
Buckets list:

Name	AWS Region	Access	Creation date
zanisebucket	Asia Pacific (Mumbai) ap-south-1	Objects can be public	June 6, 2021, 18:45:01 (UTC+01:00)

Lets upload photos to the bucket



After choosing the pics to upload click on upload



The screenshot shows the AWS S3 console interface. At the top, there's a progress bar indicating an upload is in progress, with 1% completed. Below the progress bar, the status message "Uploading" is displayed, along with metrics: Total remaining: 2 files: 8.7 MB(98.94%), Estimated time remaining: 8 minutes, Transfer rate: 19.3 KB/s. A "Cancel" button is visible in the top right corner.

Upload: status

Summary

Destination	Succeeded	Failed
s3://zanisebucket	0 files, 96.0 KB (1.06%)	0 files, 0 B (0%)

Files and folders | Configuration

Upload succeeded
View details below.

Summary

Destination	Succeeded	Failed
s3://zanisebucket	2 files, 8.8 MB (100.00%)	0 files, 0 B (0%)

Files and folders (2 Total, 8.8 MB)

Name	Folder	Type	Size	Status	Error
zani.jpg	-	image/jpeg	7.8 MB	Succeeded	-
zanis.jpg	-	image/jpeg	1.0 MB	Succeeded	-

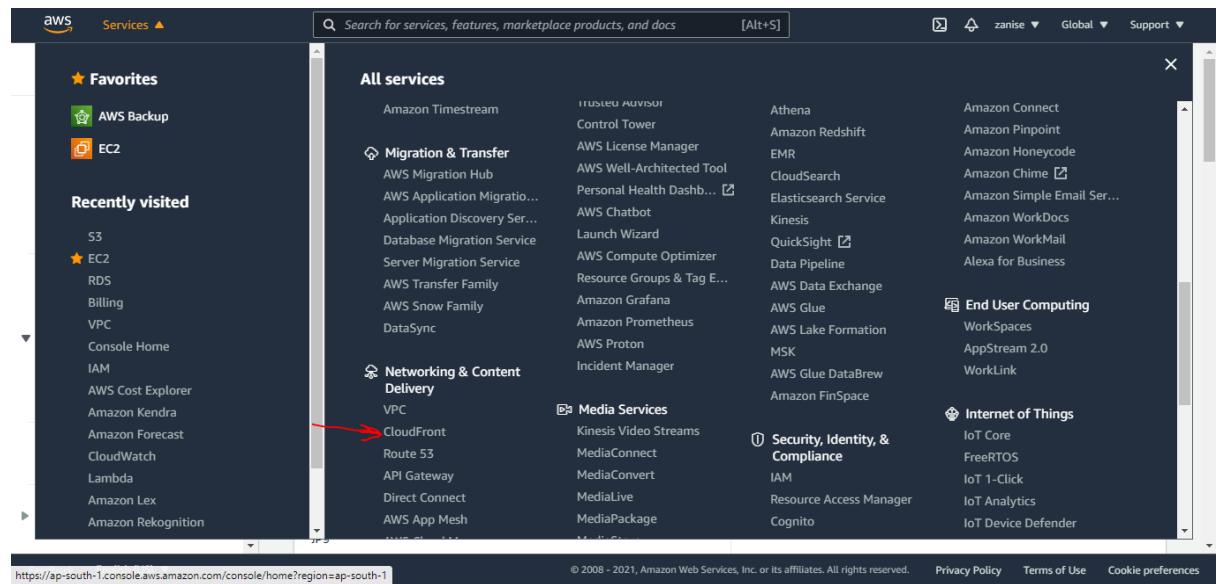
<https://zanisebucket.s3.ap-south-1.amazonaws.com/zani.jpg>

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

<https://zanisebucket.s3.ap-south-1.amazonaws.com/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": "E1WFB7ZSVKCBG",
    "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": ""
                        }
                    }
                ]
            }
        }
    }
}
```

CloudFront

Distributions Policies Functions **NEW** What's new *

▼ Telemetry Monitoring Alarms **NEW** Logs **NEW**

▼ Reports & analytics Cache statistics Popular objects Top referrers

CloudFront Distributions

Create Distribution Distribution Settings Delete Enable Disable

Delivery Method	ID	Domain Name	Comment	Origin	CNAMEs	Status	State	Last Modified
<input checked="" type="checkbox"/> Web	E1PDAA8AG2FXHQ	d1rn6bw8qxkqi1	-	zanisebuc	-	Deployed	Enabled	2021-06-06 19:35

CloudFront Distributions > E1PDAA8AG2FXHQ

General Origins and Origin Groups Behaviors Error Pages Restrictions Invalidations Tags

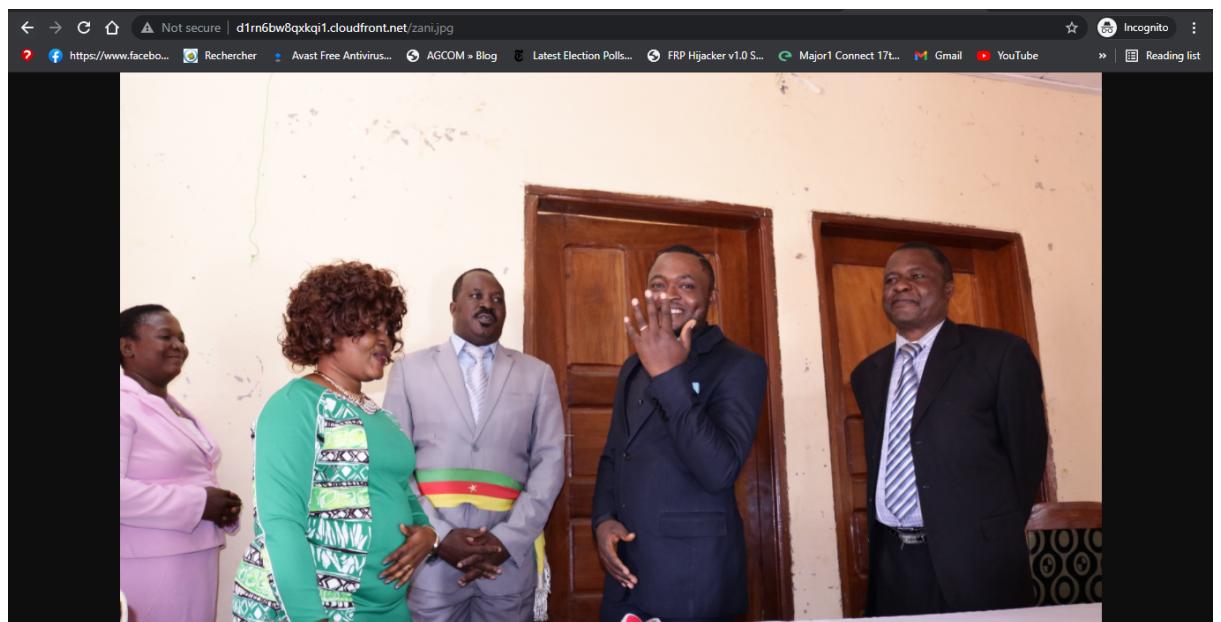
Distribution ID	E1PDAA8AG2FXHQ
ARN	arn:aws:cloudfront::497979954552:distribution/E1PDAA8AG2FXHQ
Log Prefix	-
Delivery Method	Web
Cookie Logging	Off
Distribution Status	Deployed
Comment	-
Price Class	Use All Edge Locations (Best Performance)
AWS WAF Web ACL	-
State	Enabled
Alternate Domain Names (CNAMEs)	-
SSL Certificate	Default CloudFront Certificate (*.cloudfront.net)
Domain Name	d1rn6bw8qxkqi1.cloudfront.net
Custom SSL Client Support	-
Security Policy	TLSv1
Supported HTTP Versions	HTTP/2, HTTP/1.1, HTTP/1.0
IPv6	Enabled
Default Root Object	-
Last Modified	2021-06-06 19:35 UTC+1
Log Bucket	-

d1rn6bw8qxkqi1.cloudfront.net/zani.jpg

d1rn6bw8qxkqi1.cloudfront.net/zanis.jpg

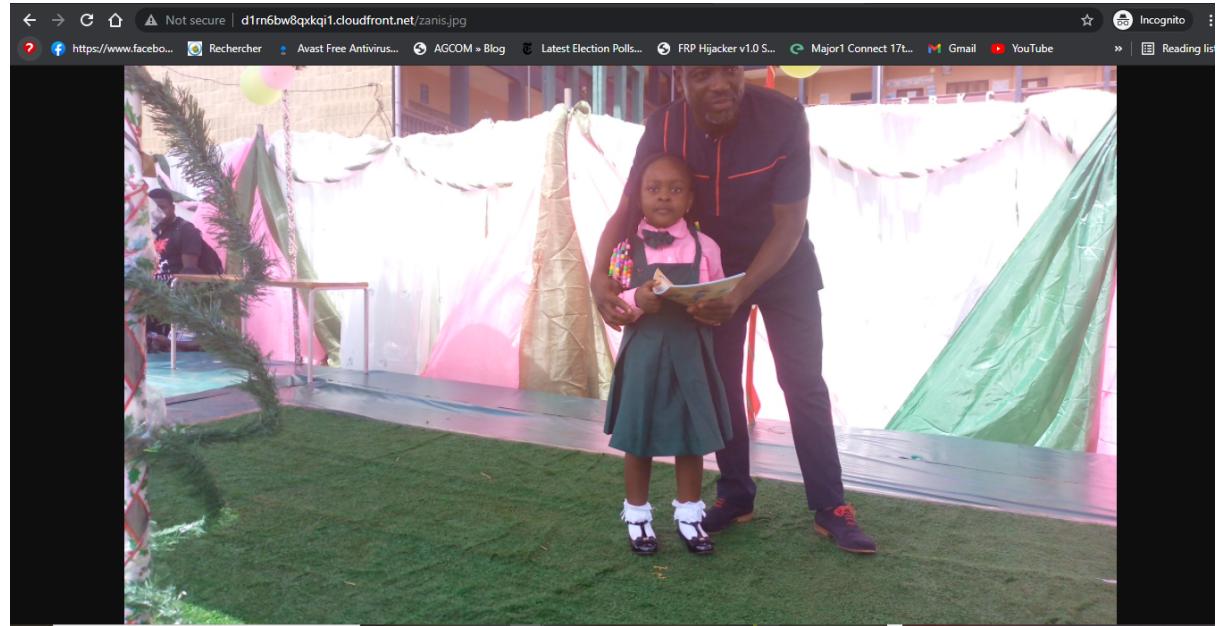
lets test and see if it works

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.

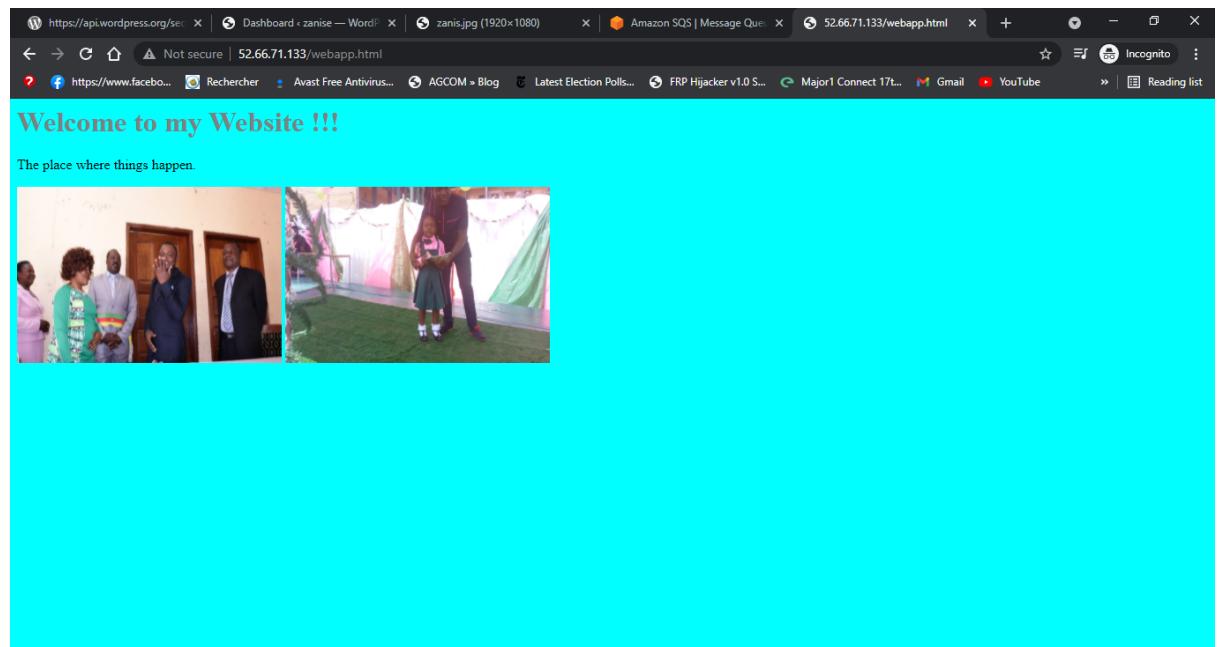
```
root@ip-172-31-46-149:/var/www/html
<body background="aqua">
<h1 style="color:grey">Welcome to my Website !!!</h1>
<p>The place where things happen.</p>


</body>
~
```

Lets test to see if it works

using the instance IP and the webapp.html

<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 #awscesa2020

#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](#)