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

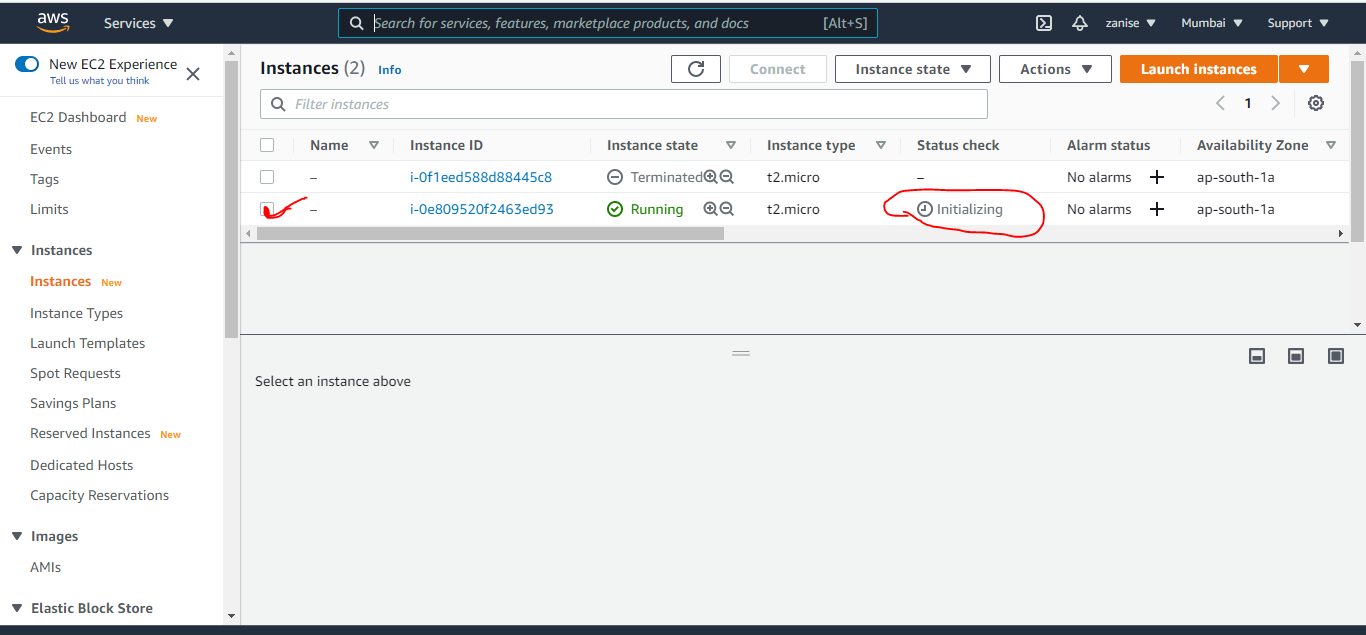
The architecture includes -

- Webserver configured on EC2 Instance

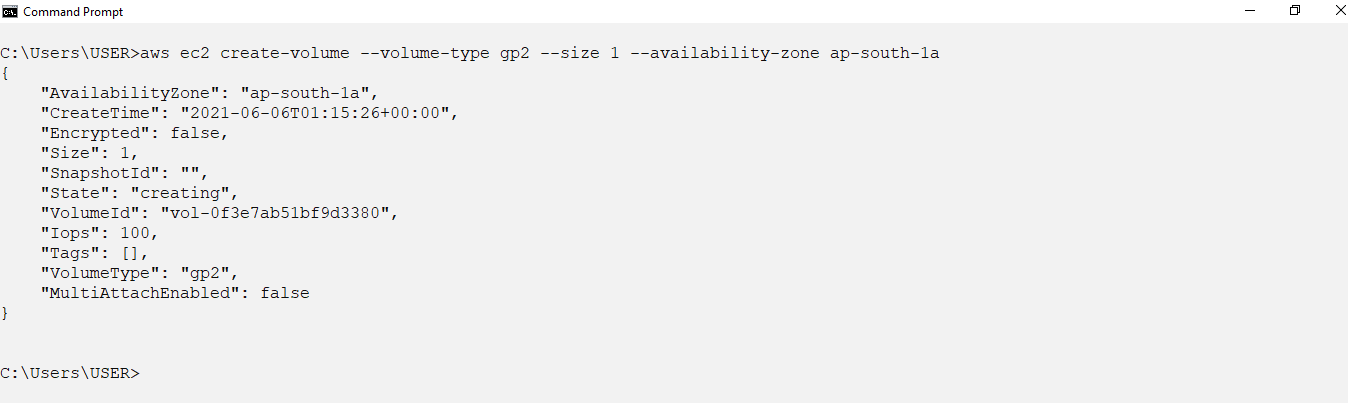
- Document Root(/var/www/html) made persistent by mounting on EBS Block Device.

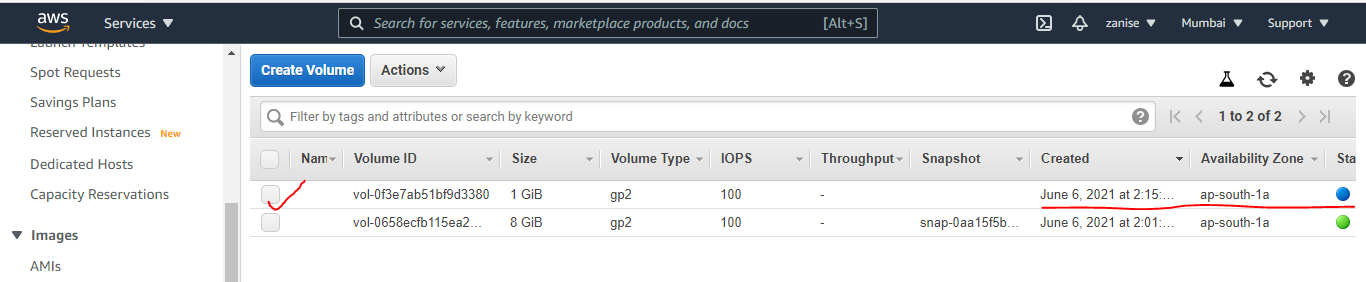
1. lets create an ec2 instance





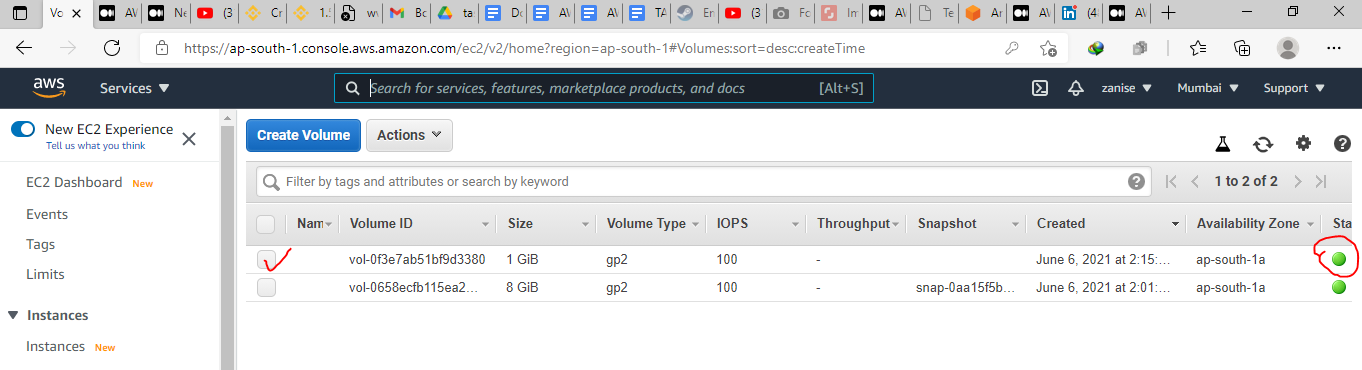
2. lets create a volume (EBS)





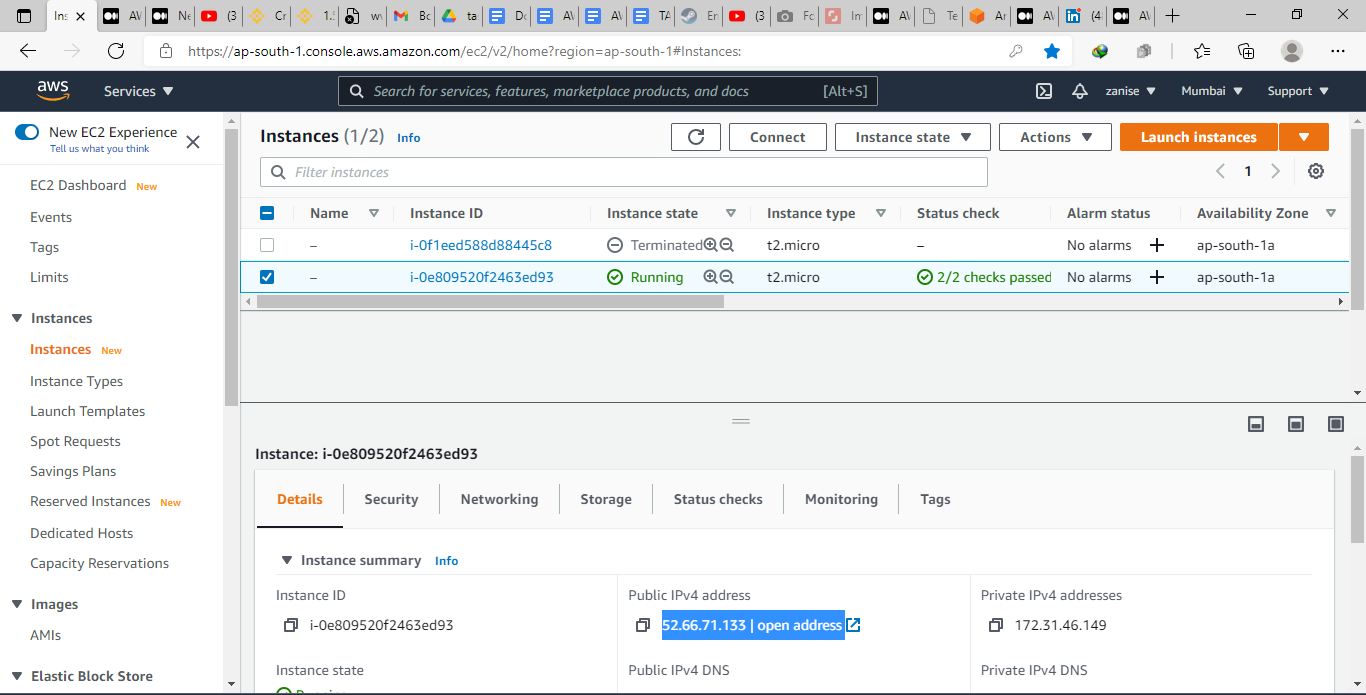
3. Attaching EBS to EC2

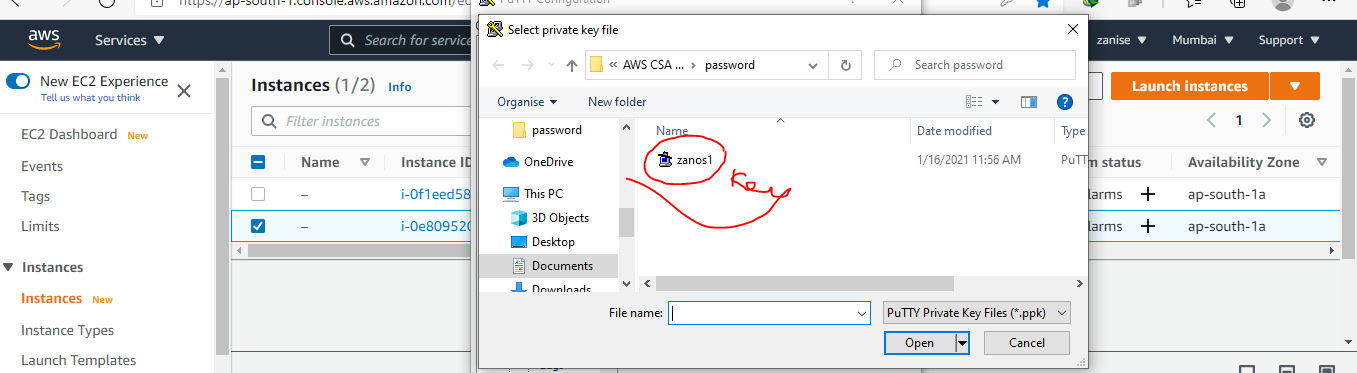


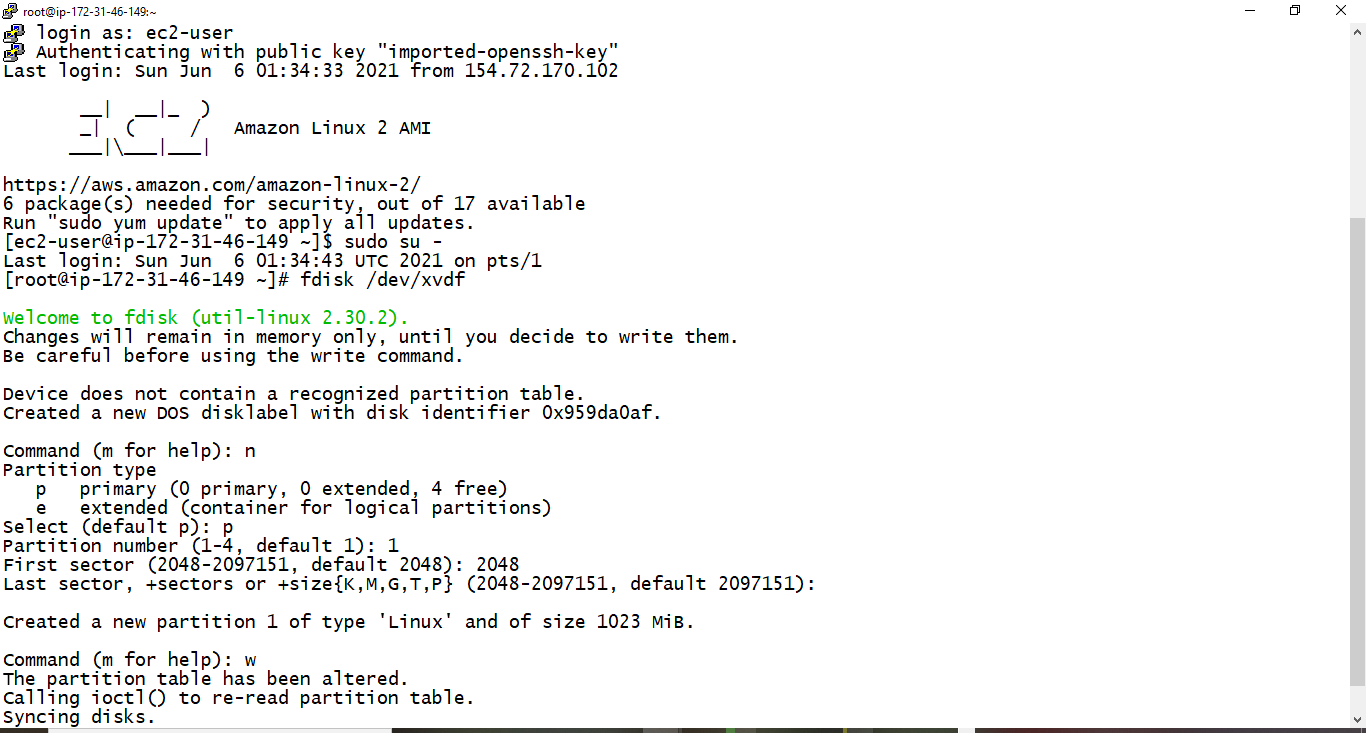


4- Formating EBS partition for that we will need to sshinto the instance using putty. for that we need the instane

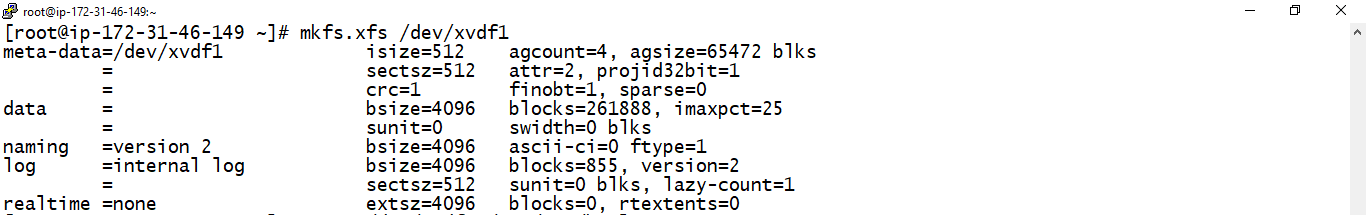
* Public IPv4
* private key



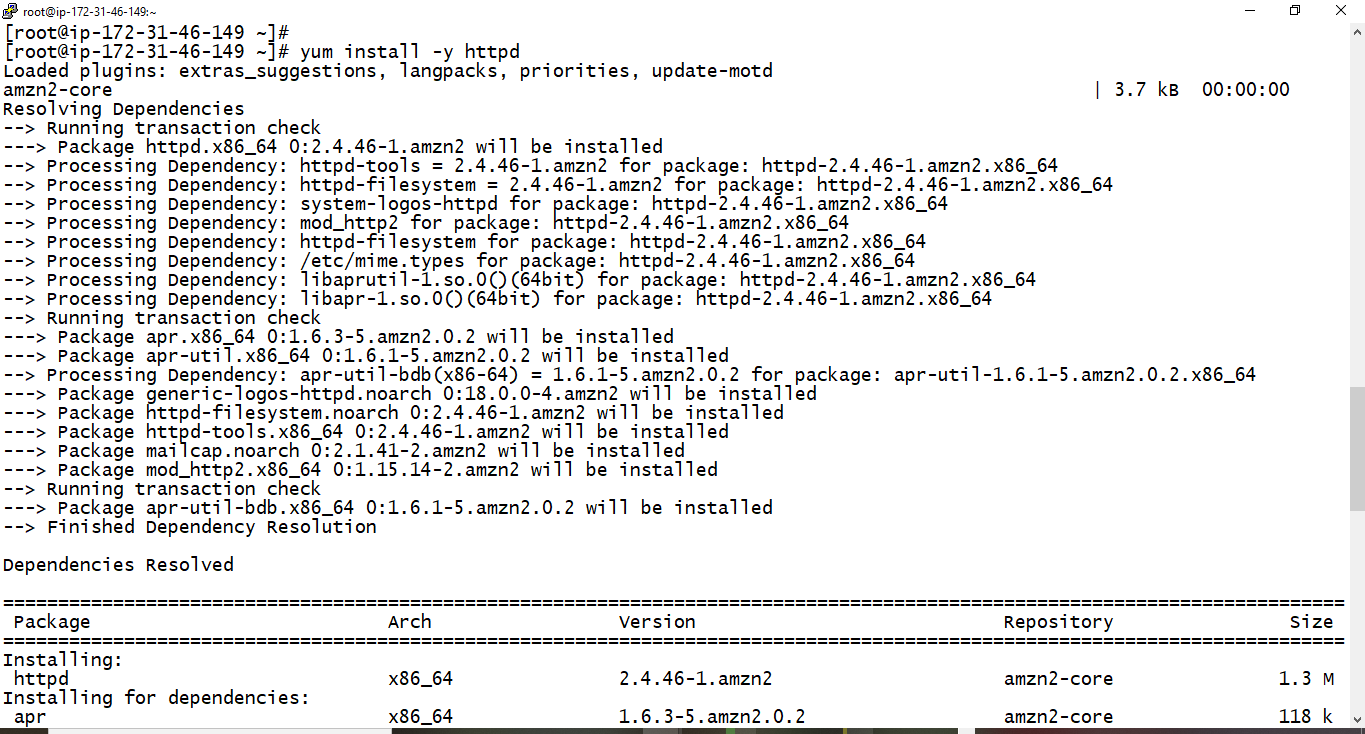


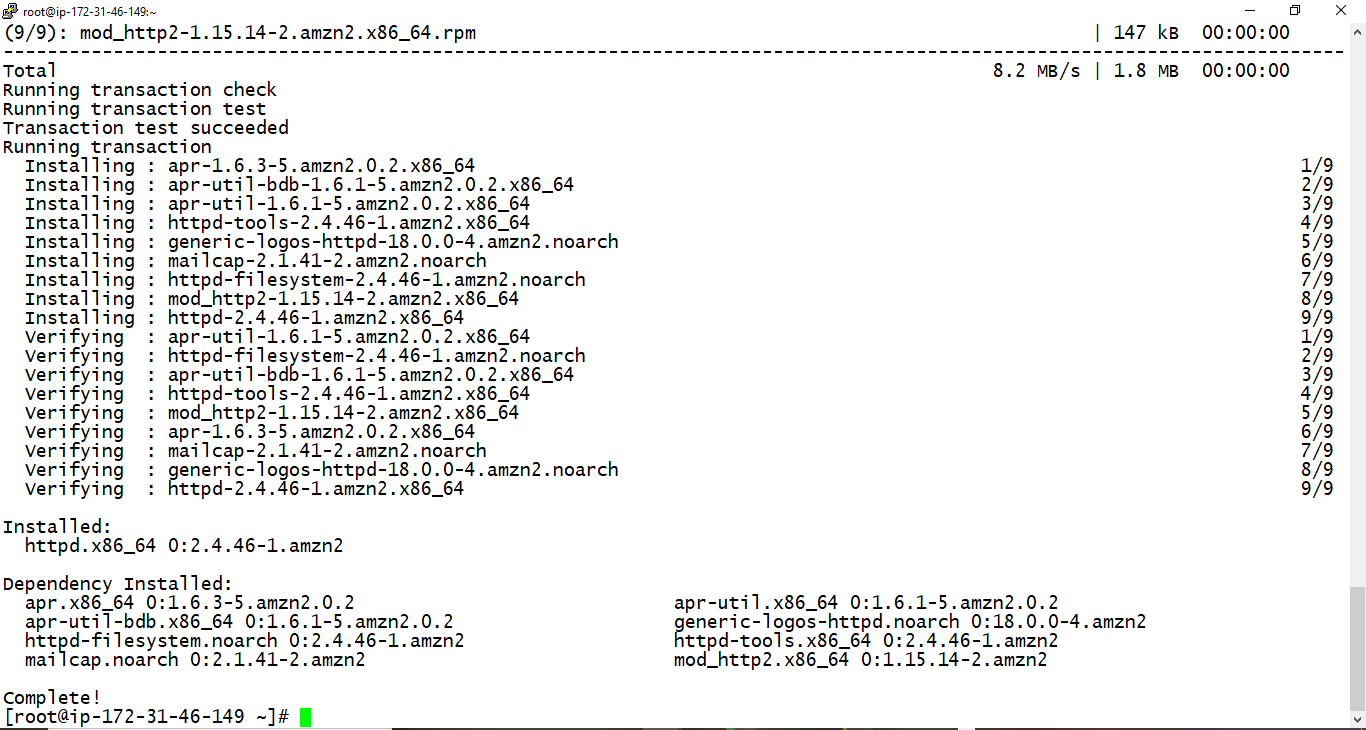


5. lets format the partition created

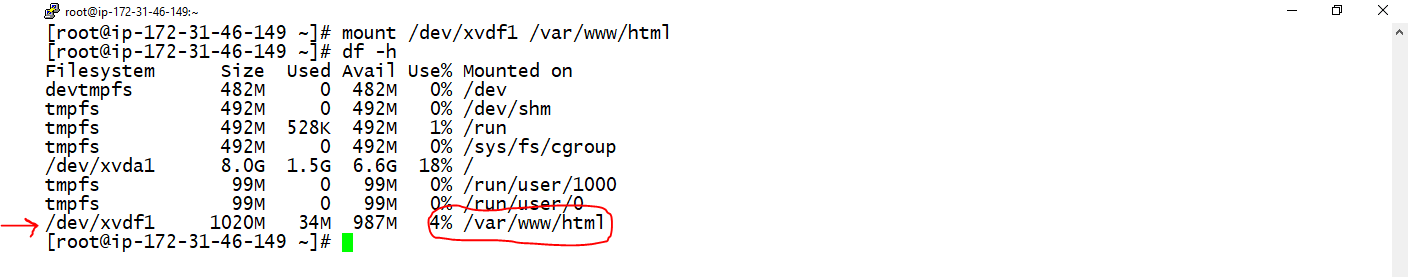


6. lets install apache webserver

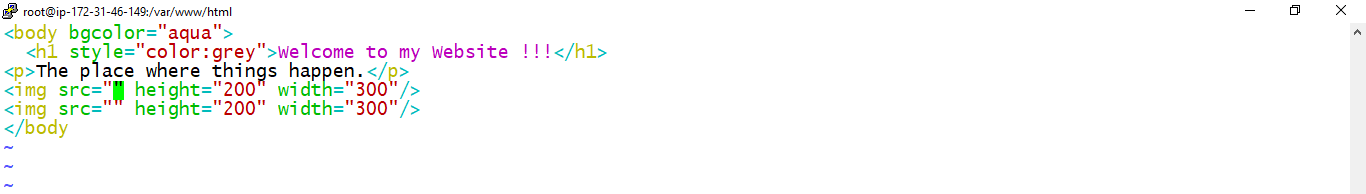




7. lets mount /dev/xvdf1 on /var/www/html

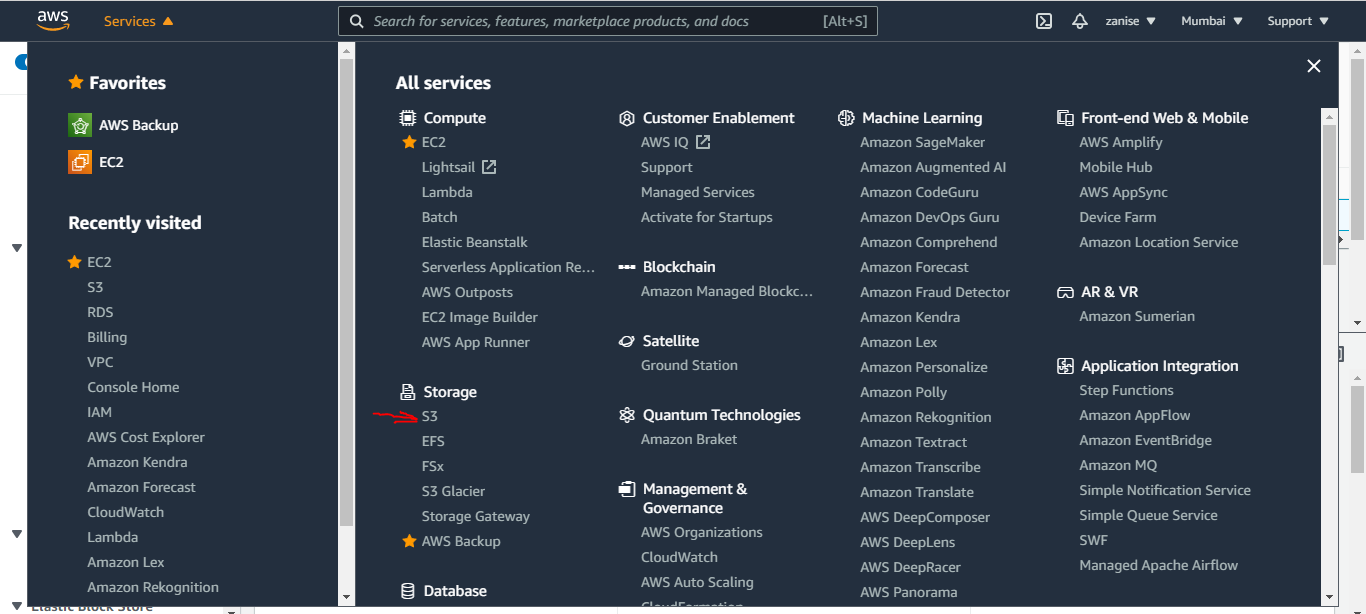


8. Writing a webapp code in the /var/www/html folder

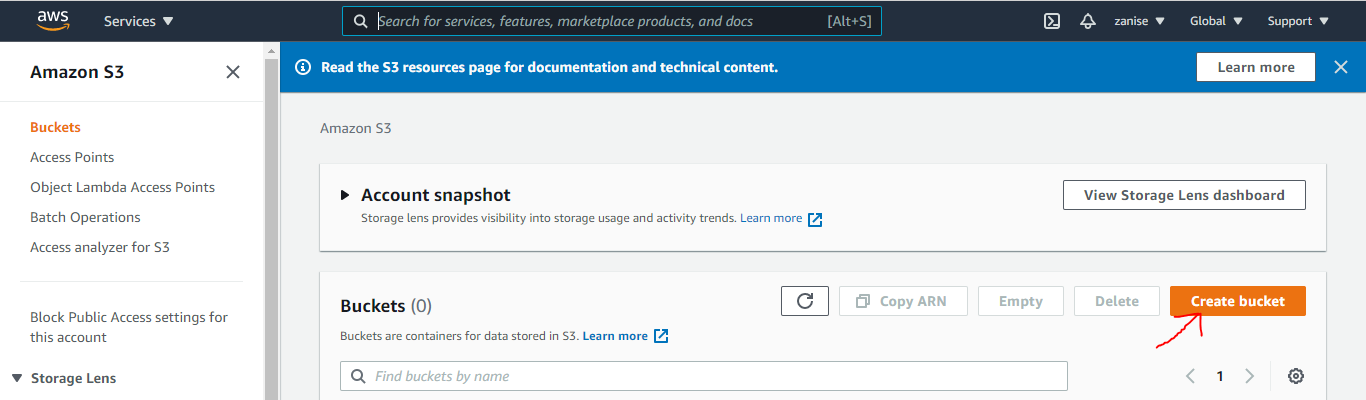


9- Lets create an s3 bucket then upload pictures there

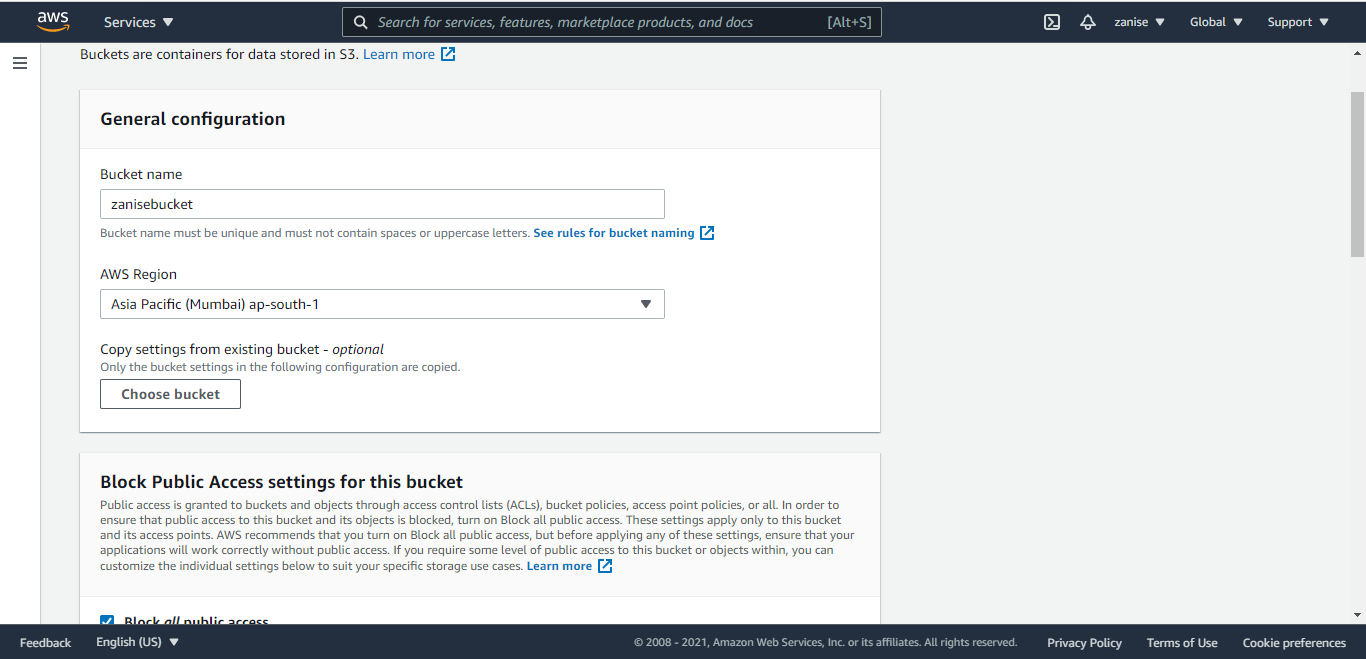
go to service then click on s3

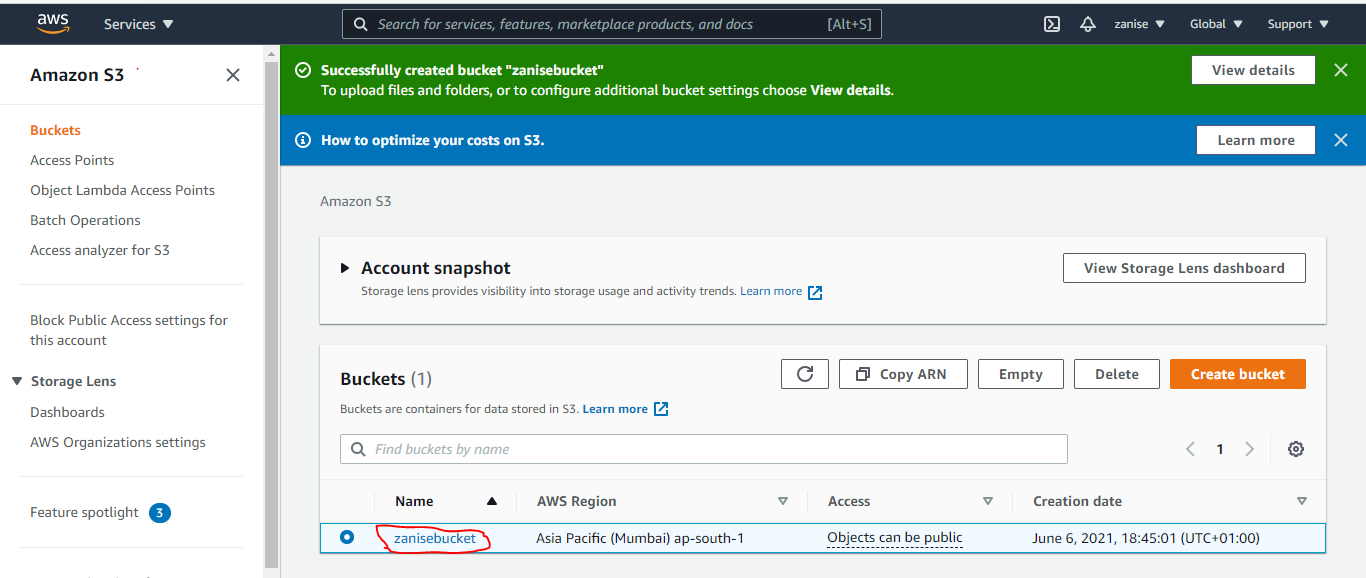


click on create bucket

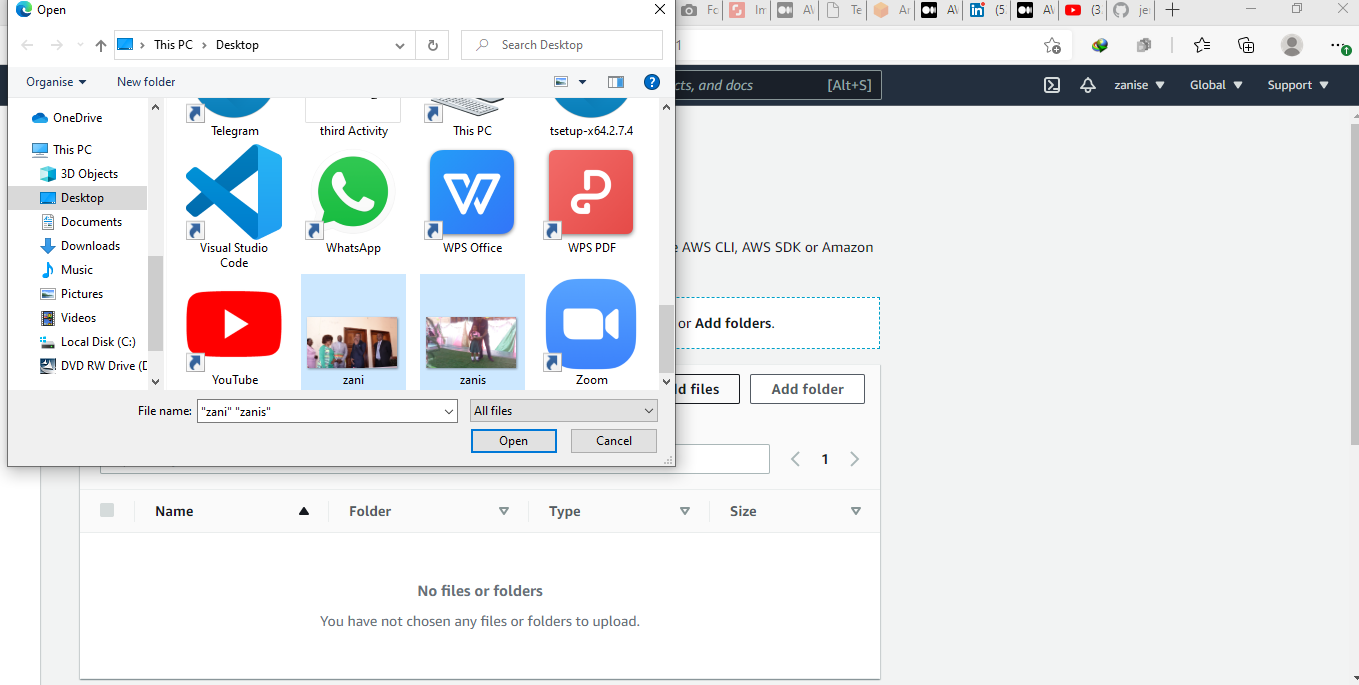


fill the bucket name and the aws region that is all then click on create bucket

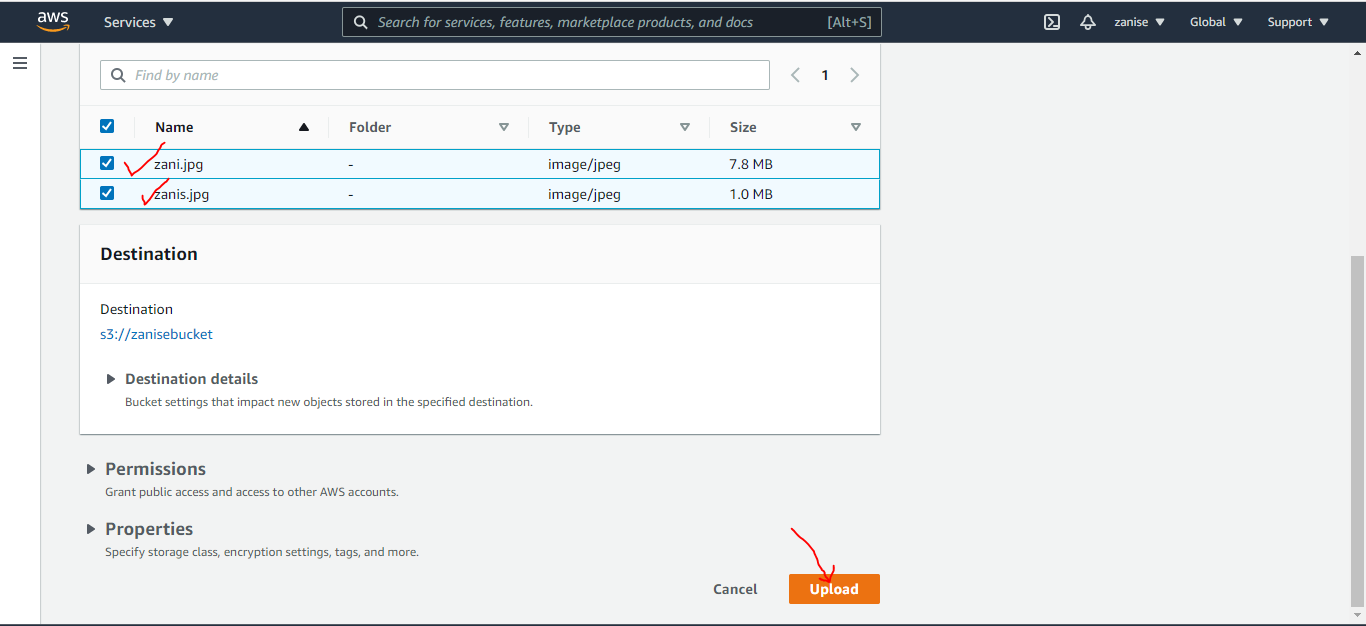


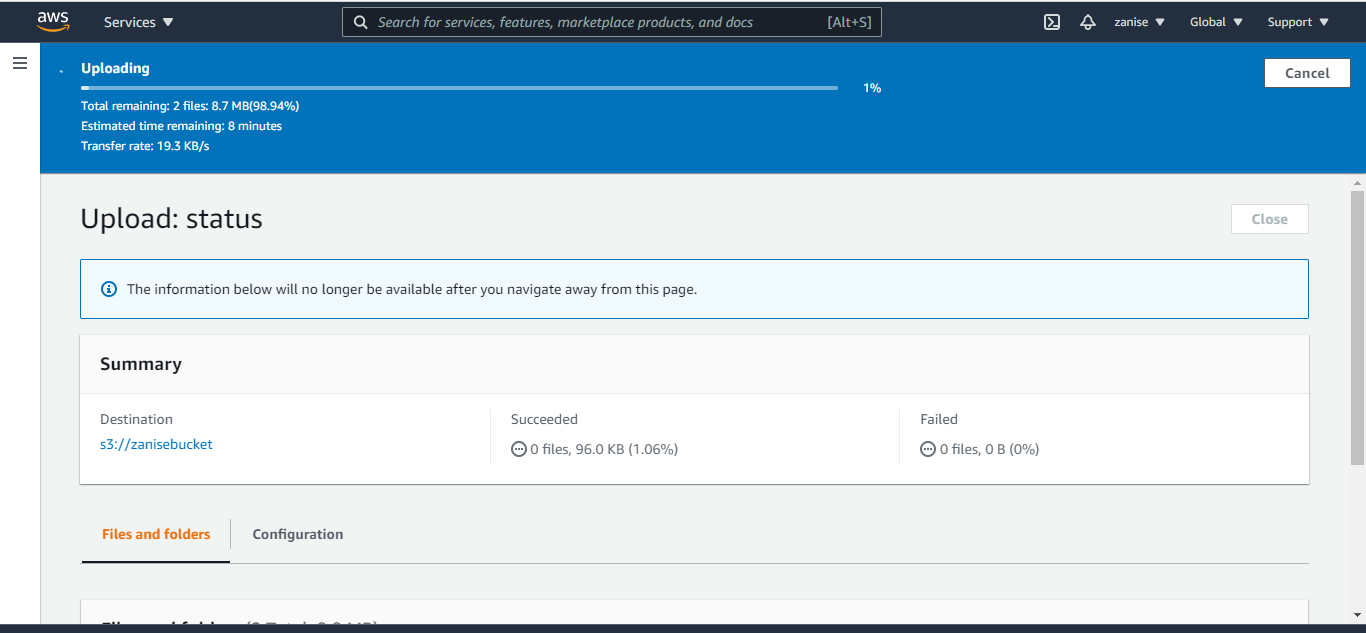


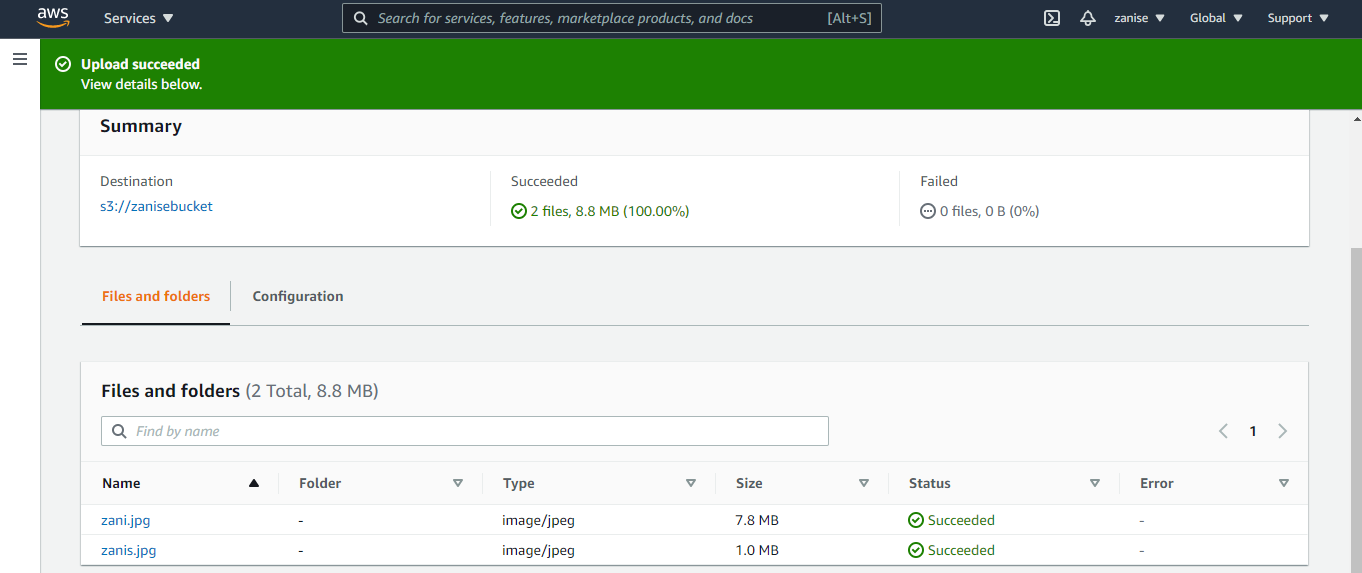
Lets upload photos to the bucket



After choosing the pics to upload click on upload



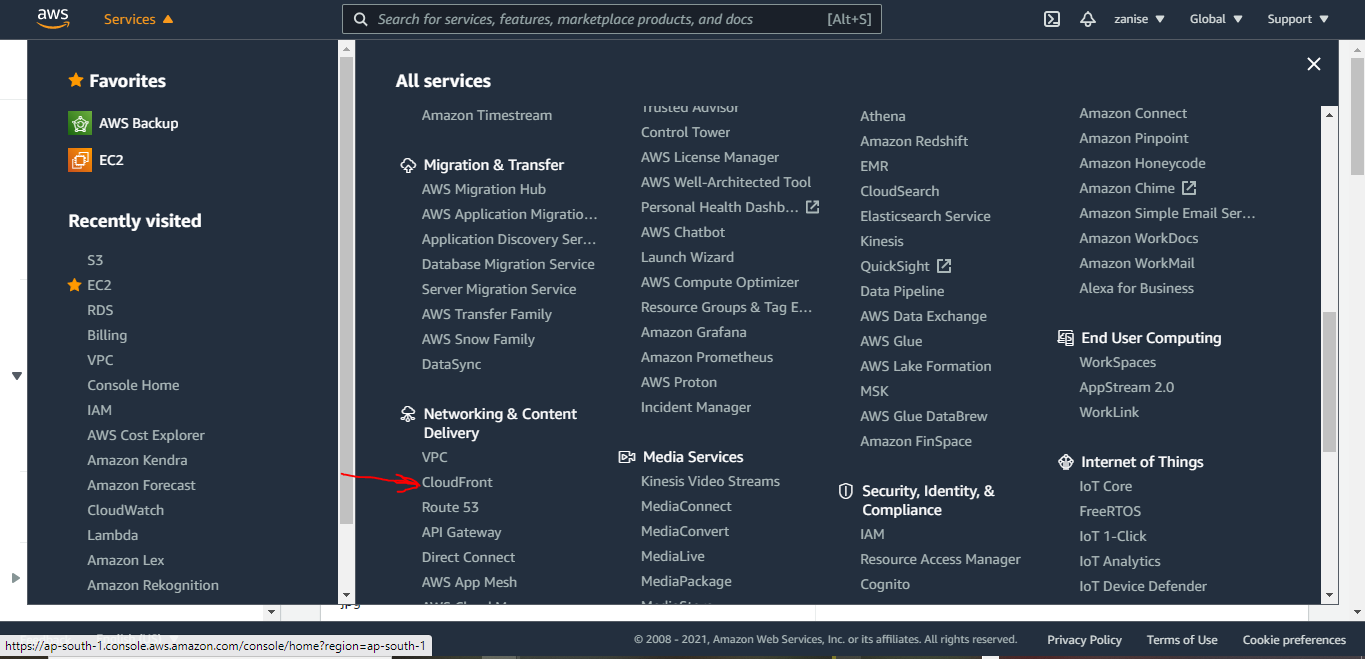




<https://zanisebucket.s3.ap-south-1.amazonaws.com/zani.jpg> this the url for zani.jpg

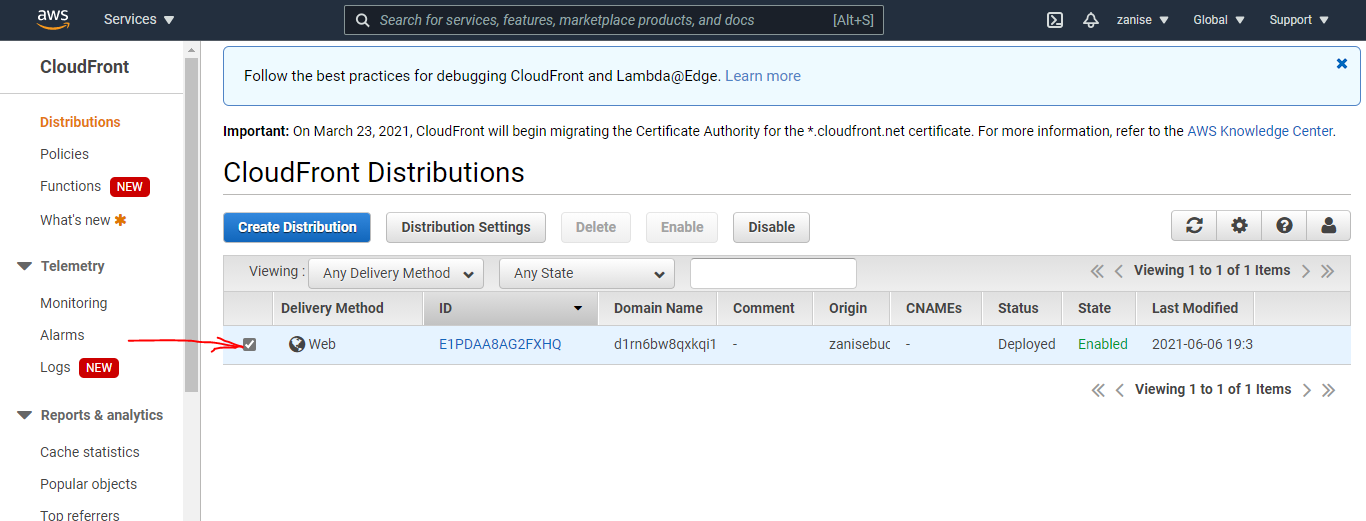
<https://zanisebucket.s3.ap-south-1.amazonaws.com/zanis.jpg> this the url for zanis.jpg

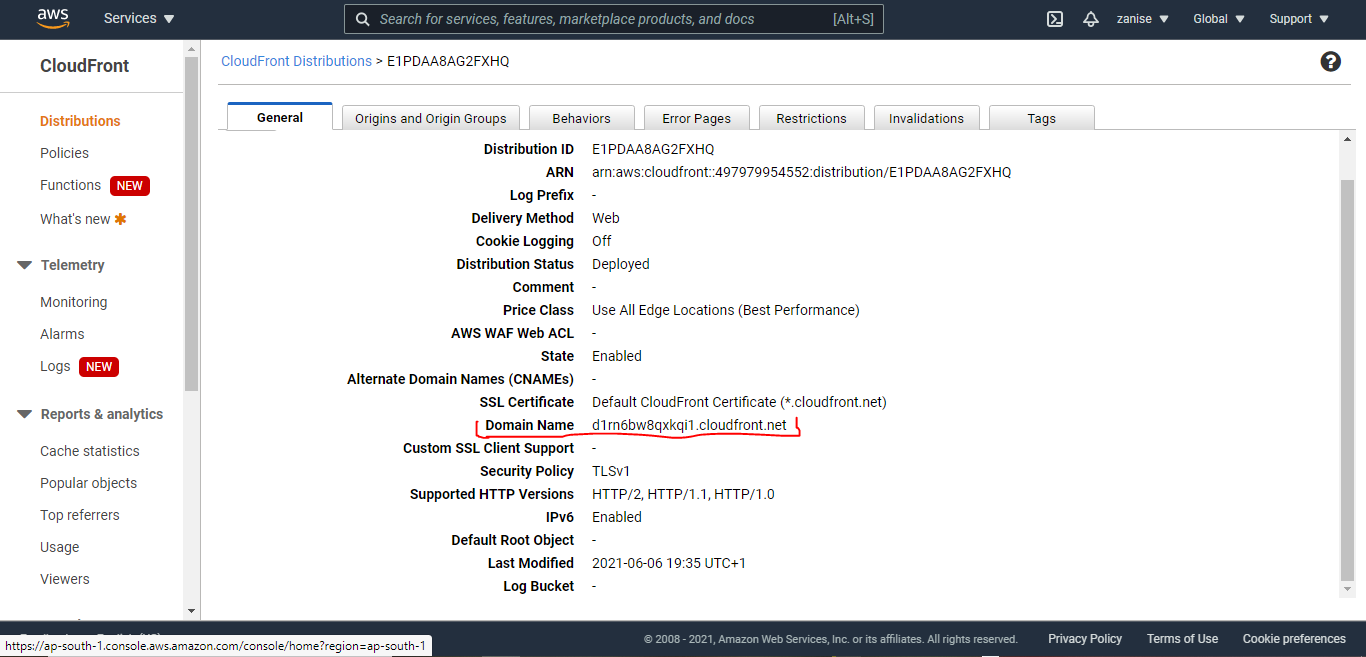
10. Setting up Content Delivery Network using CloudFront and using the origin domain as S3 bucket



lets create a distribution





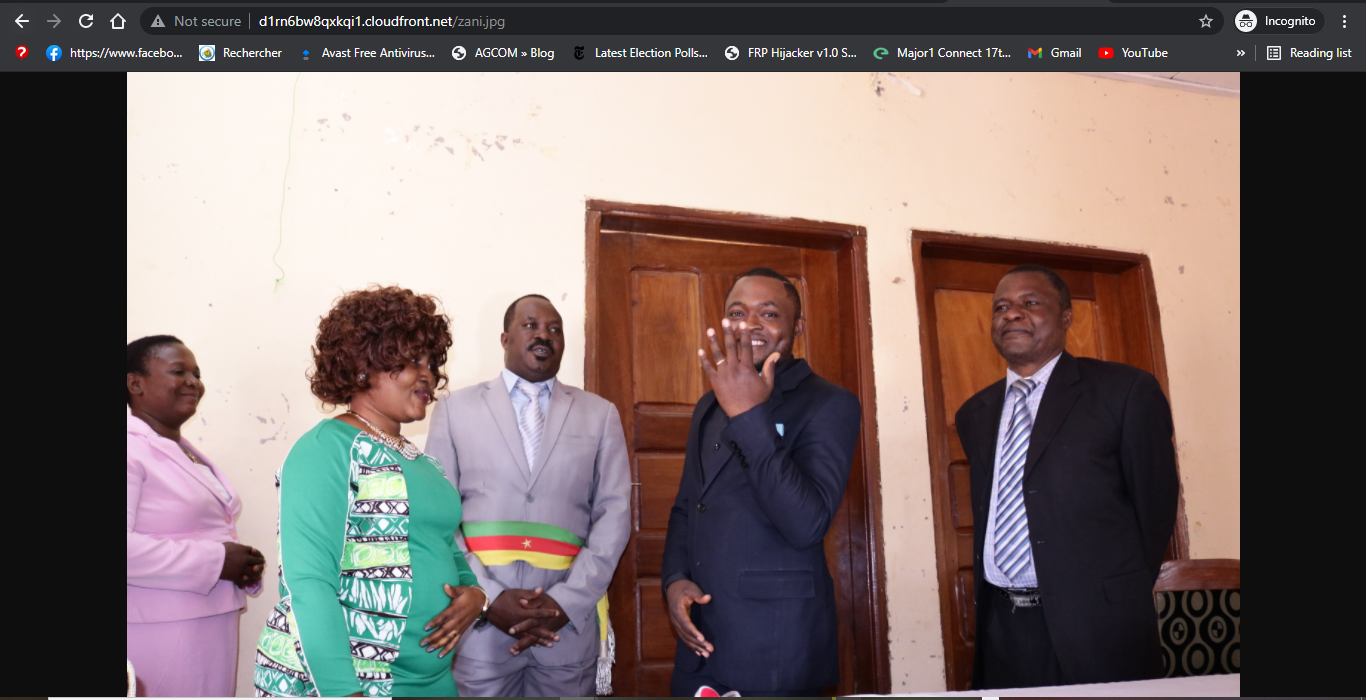


d1rn6bw8qxkqi1.cloudfront.net/zani.jpg

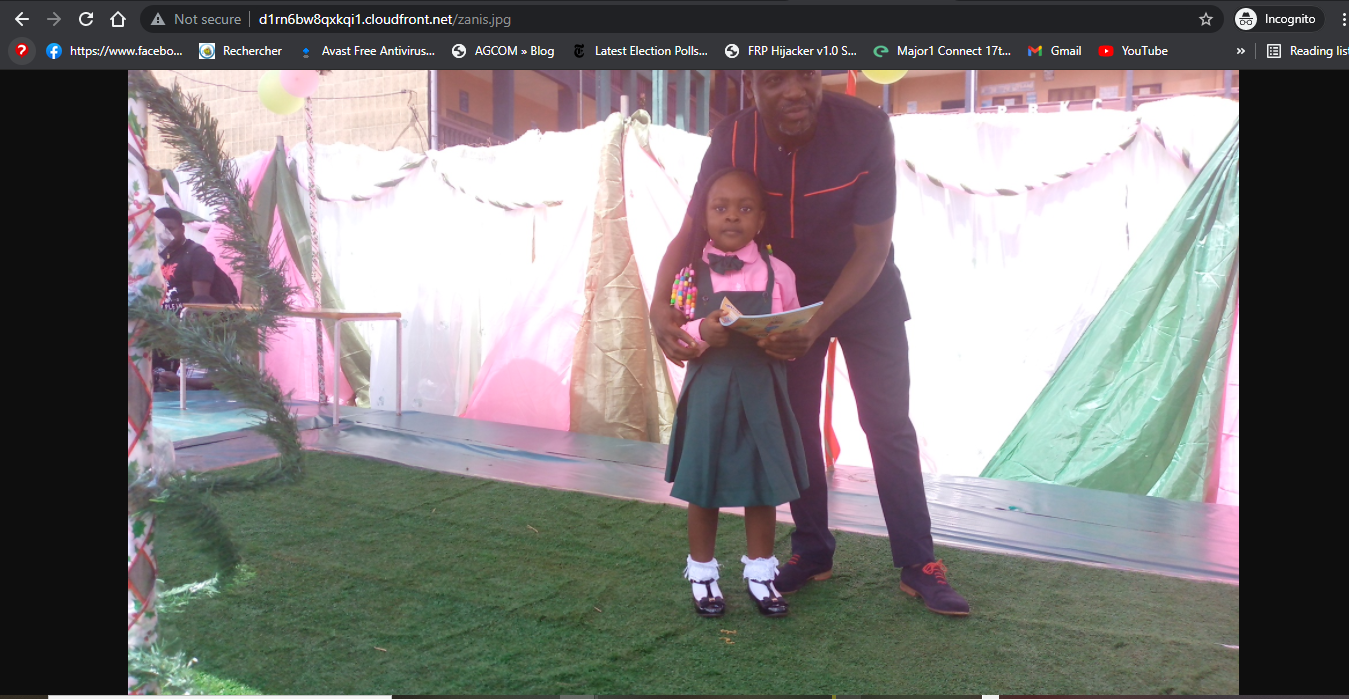
d1rn6bw8qxkqi1.cloudfront.net/zanis.jpg

lets test and see if it works

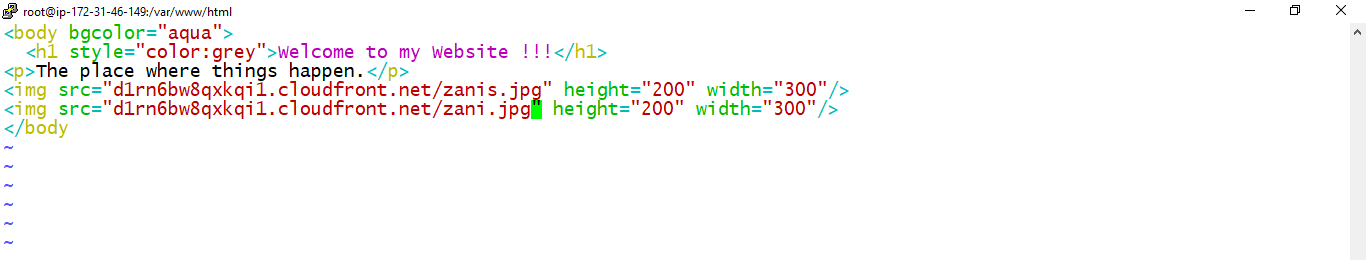
1. https://d1rn6bw8qxkqi1.cloudfront.net/zani.jpg



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



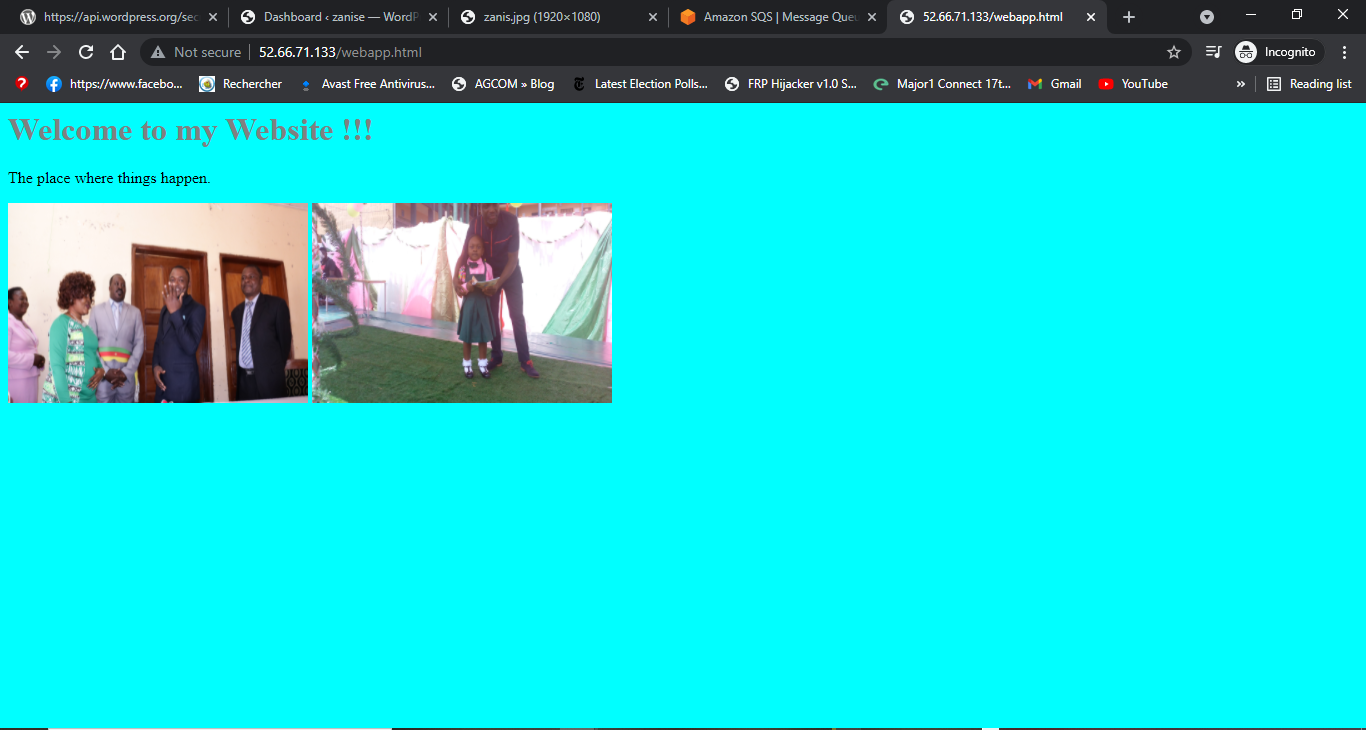
11. So lets Finally place the Cloud Front URL on the webapp code for security and low latency.



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](https://kzanise.medium.com/?source=post_sidebar--------------------------post_sidebar-----------)

Am a young passionate of IT

## [More from zanise khan](https://kzanise.medium.com/?source=follow_footer-------------------------------------)

Am a young passionate of IT

[About](https://medium.com/about?autoplay=1&source=post_page-----d58967e16173--------------------------------)

[Help](https://help.medium.com/hc/en-us?source=post_page-----d58967e16173--------------------------------)

[Legal](https://policy.medium.com/medium-terms-of-service-9db0094a1e0f?source=post_page-----d58967e16173--------------------------------)