**29.AWS-B30-S3-Part-1**

--- in this session, we will discuss about s3 (simple storage service)

--- s3 is an object storage. No storage limit for s3 which why we call it as bucket.

--- we can store any type of files like logs, mp4, dat, backup…etc.

--- max size of a file we can upload to s3 is 5TB.

--- s3 bucket name max length is between 3 to 63 characters. We can use (.) and (-)

**Use case of aws s3 bucket**

--- terraform state file can be saved in s3 and enable versioning.

--- to host websites on s3.

--- save logs files and move to glacier.

--- for handling huge amount of data like data lake, DWH.

**AMAZON s3 storage classes**

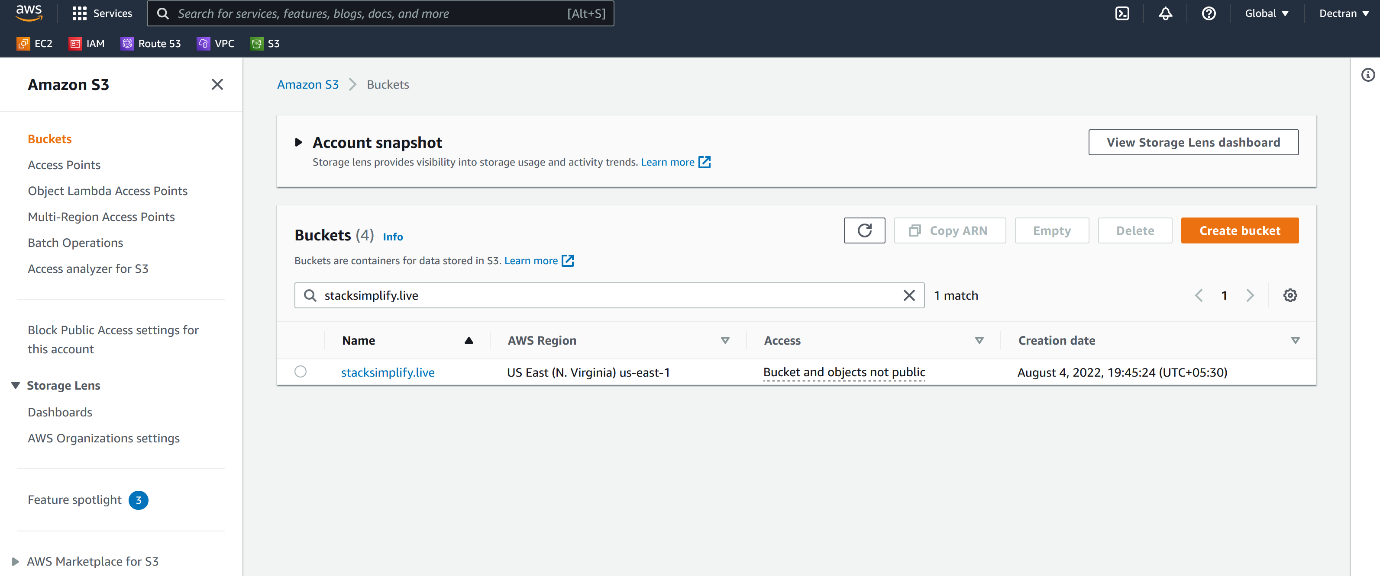
--- we have 2 things in s3

1. Availability
2. Durability

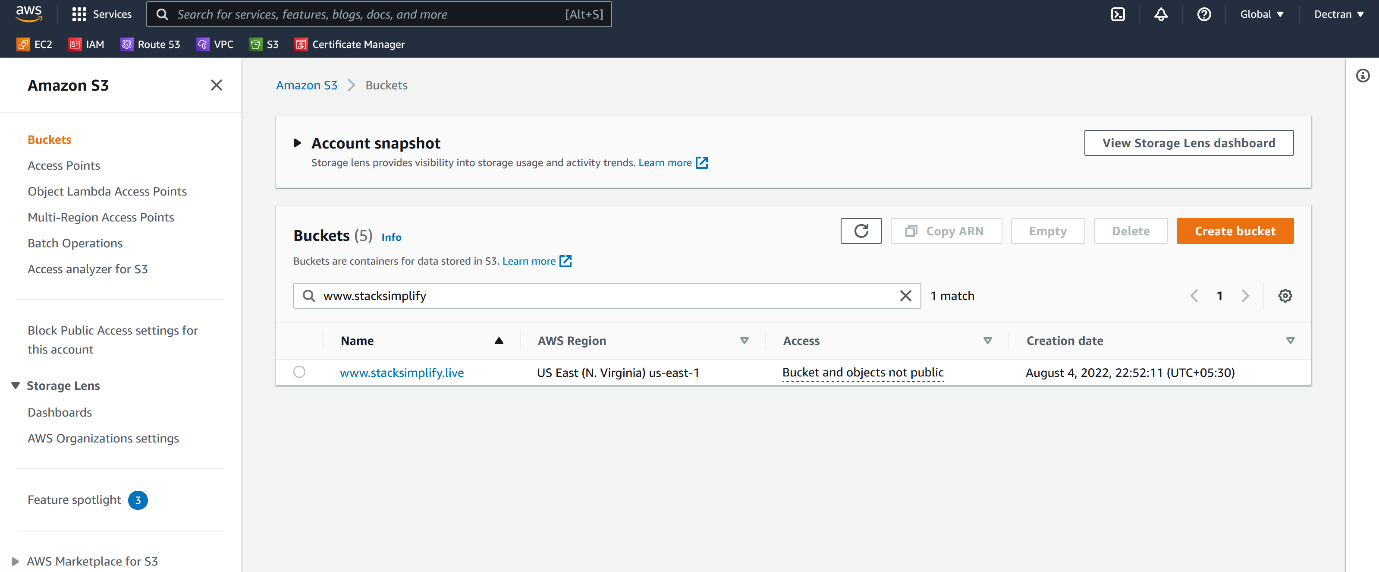
---

**Host a website on s3 bucket**

--- **note** – the website name should be same as bucket name.



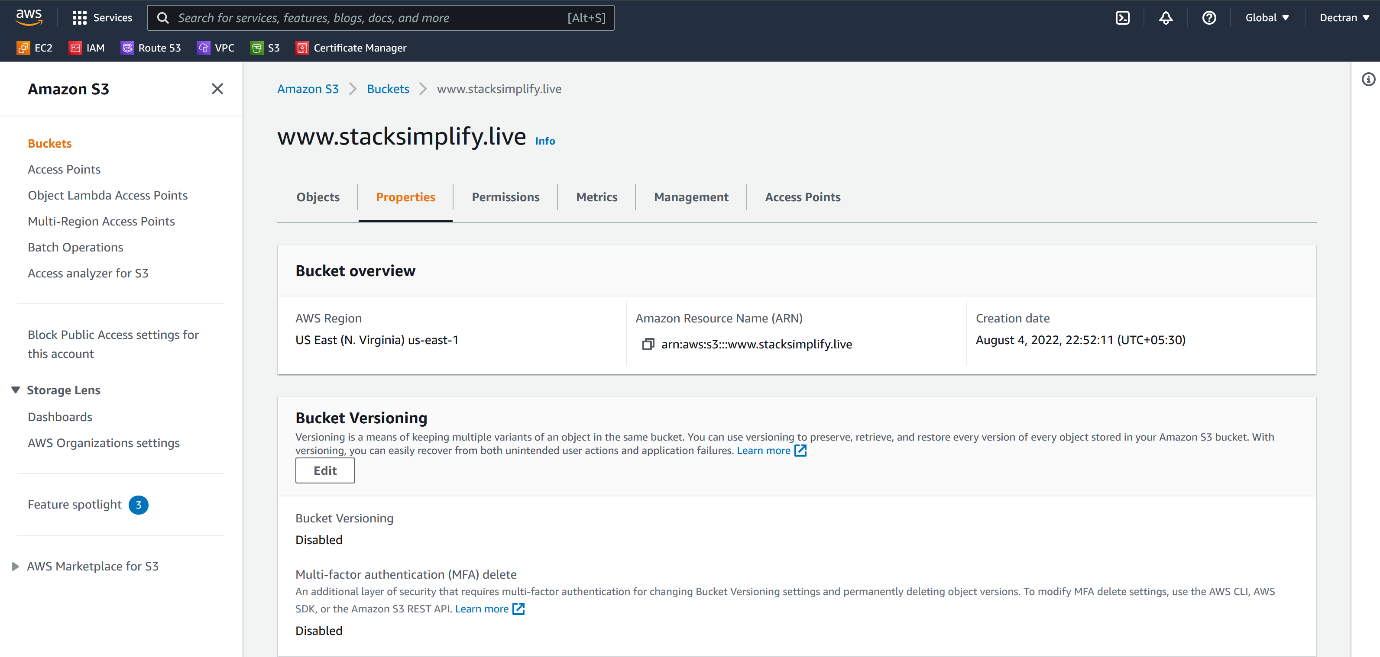
--- **note** - I have created a s3 bucket with same name as domain.



--- **note** – now users access the url [www.stacksimplify.live](http://www.stacksimplify.live) then that request will route to this bucket.

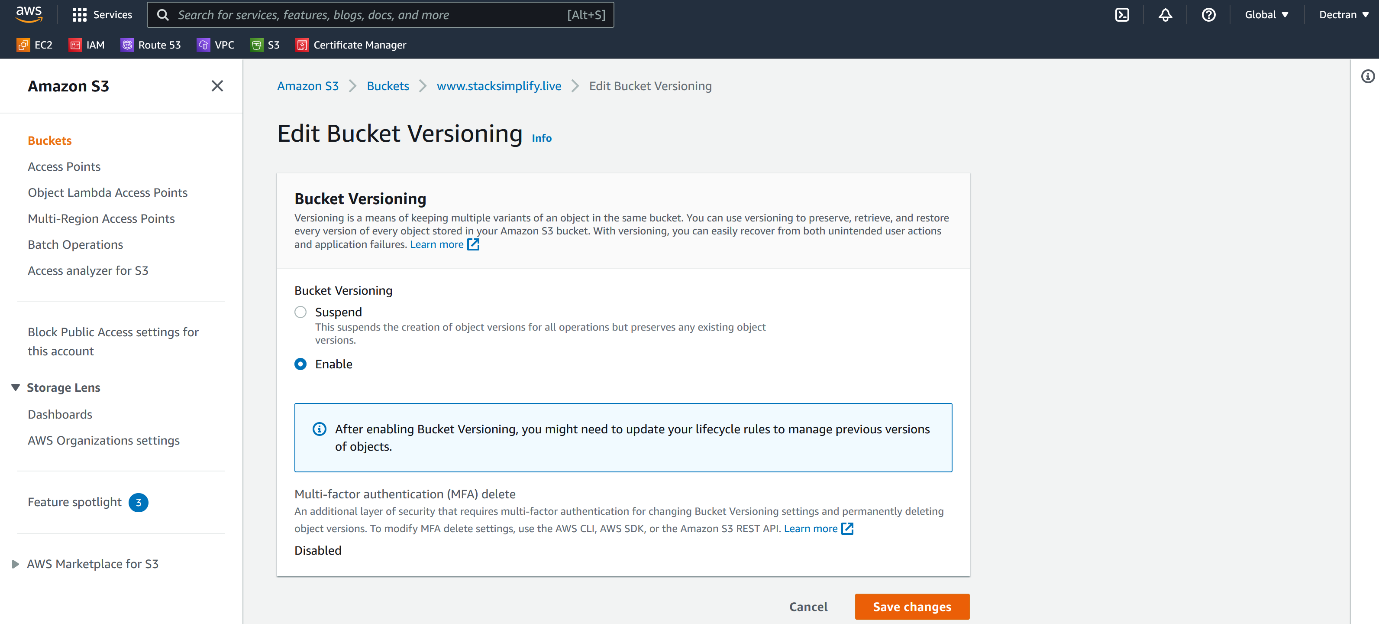
**Properties**

**Bucket versioning**



--- **Bucket Versioning** – the data inside of s3 is very important, you are storing terraform state file then we must enable versioning.

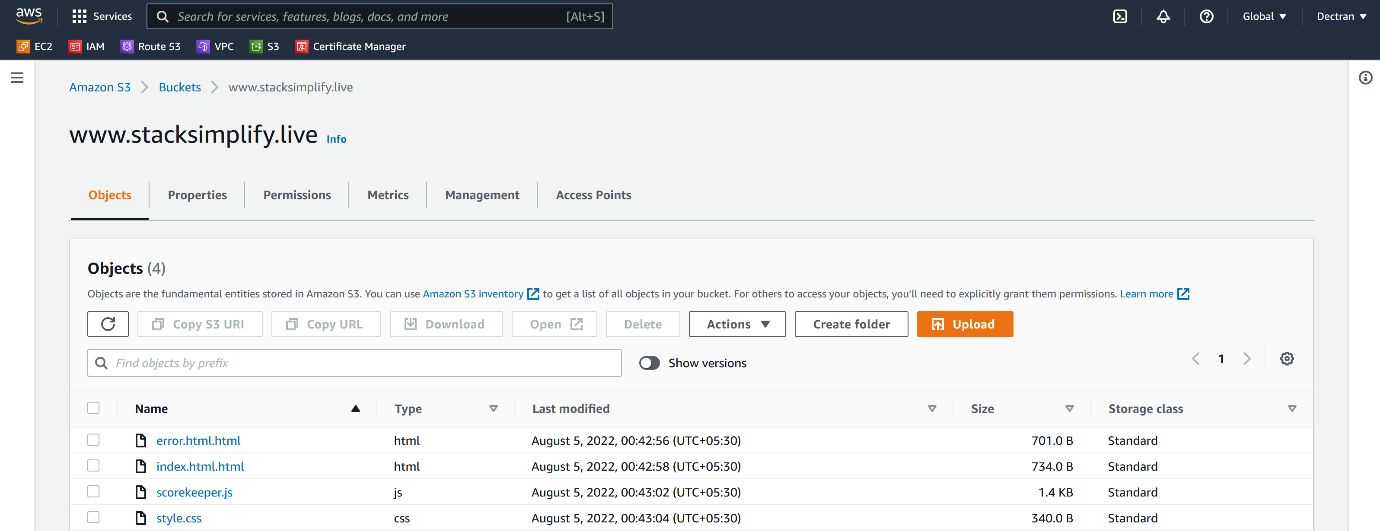
--- click on edit.



--- click on enable and save changes.

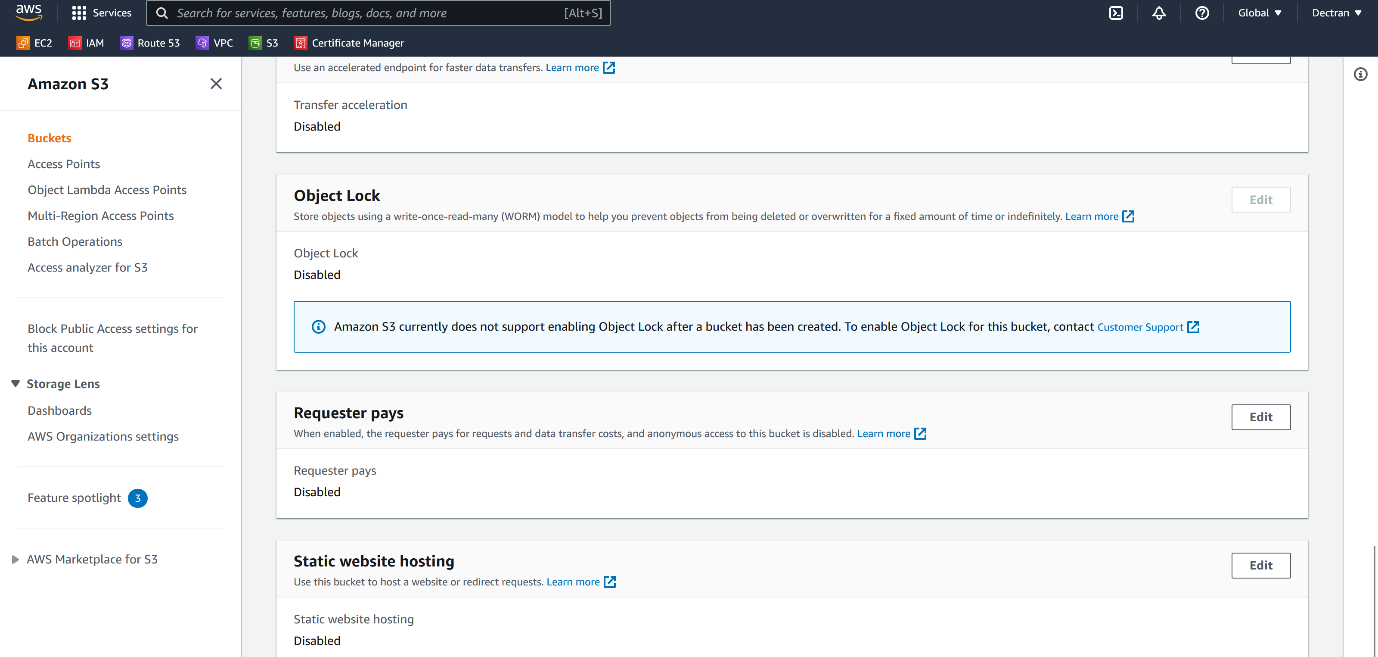
**Load website to s3 bucket**

--- Reference - <https://github.com/prabhu9652/dockertest1>

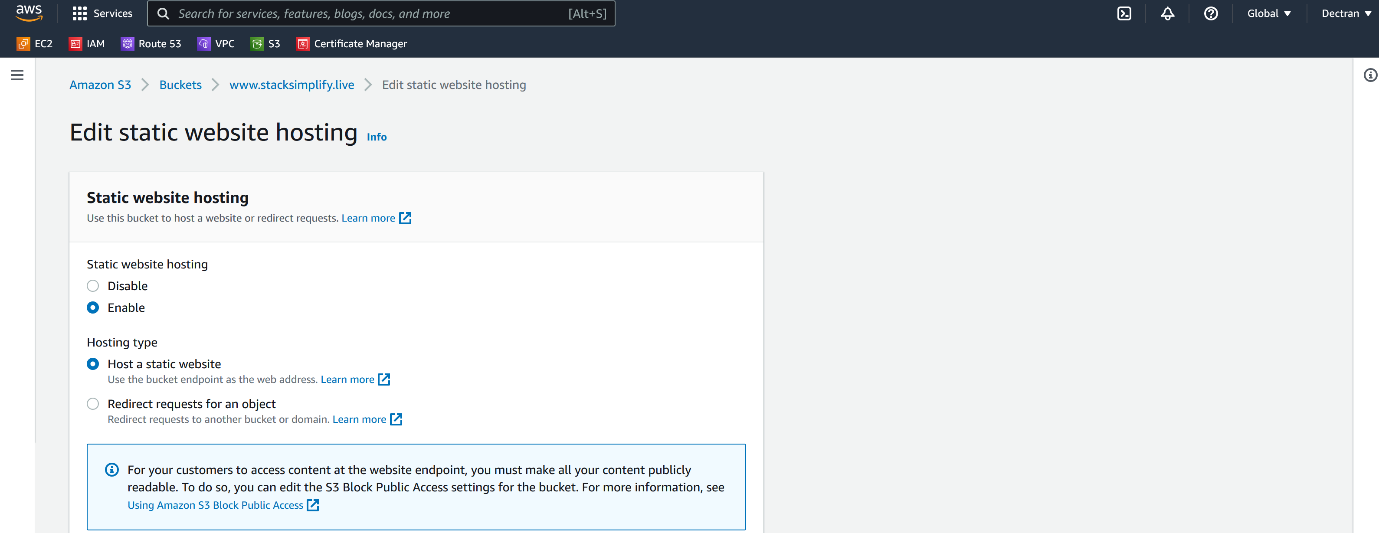


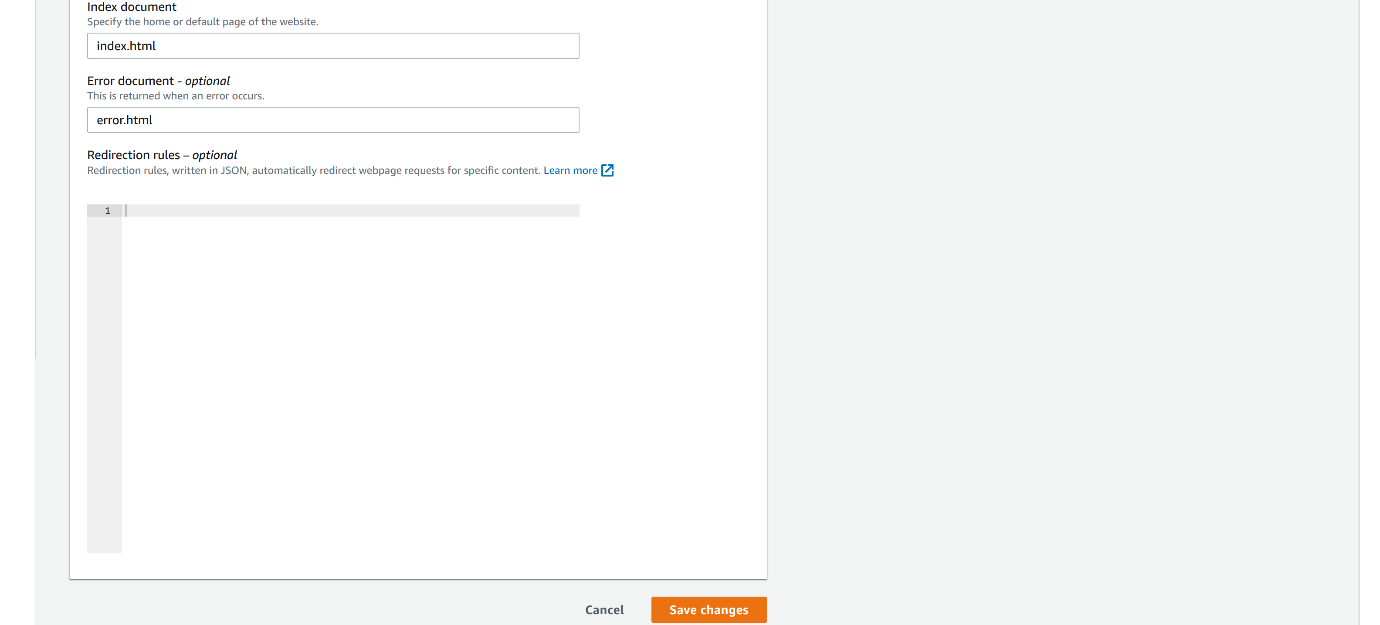
--- I have uploaded the website in aws s3.

**Static website hosting**

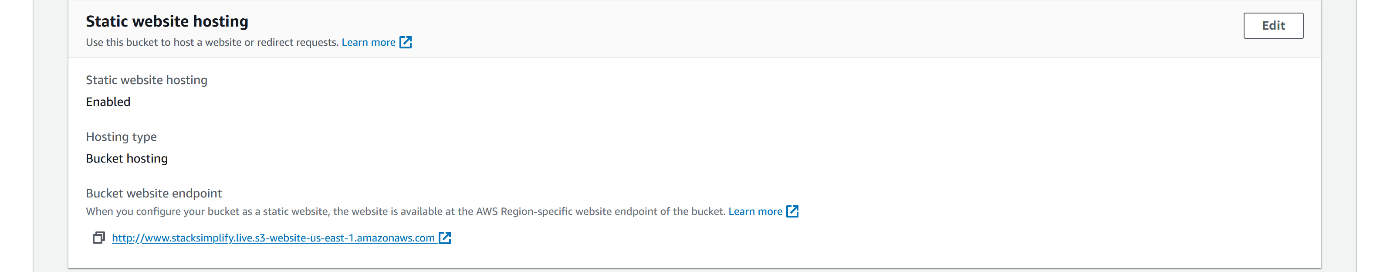


--- scroll down, you will find static website hosting.

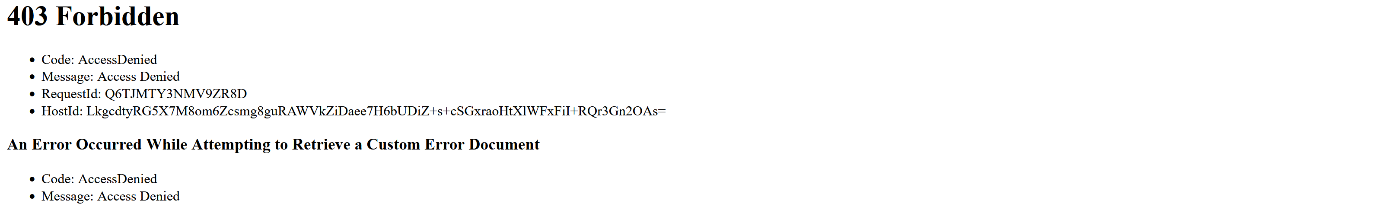




--- click on save changes.

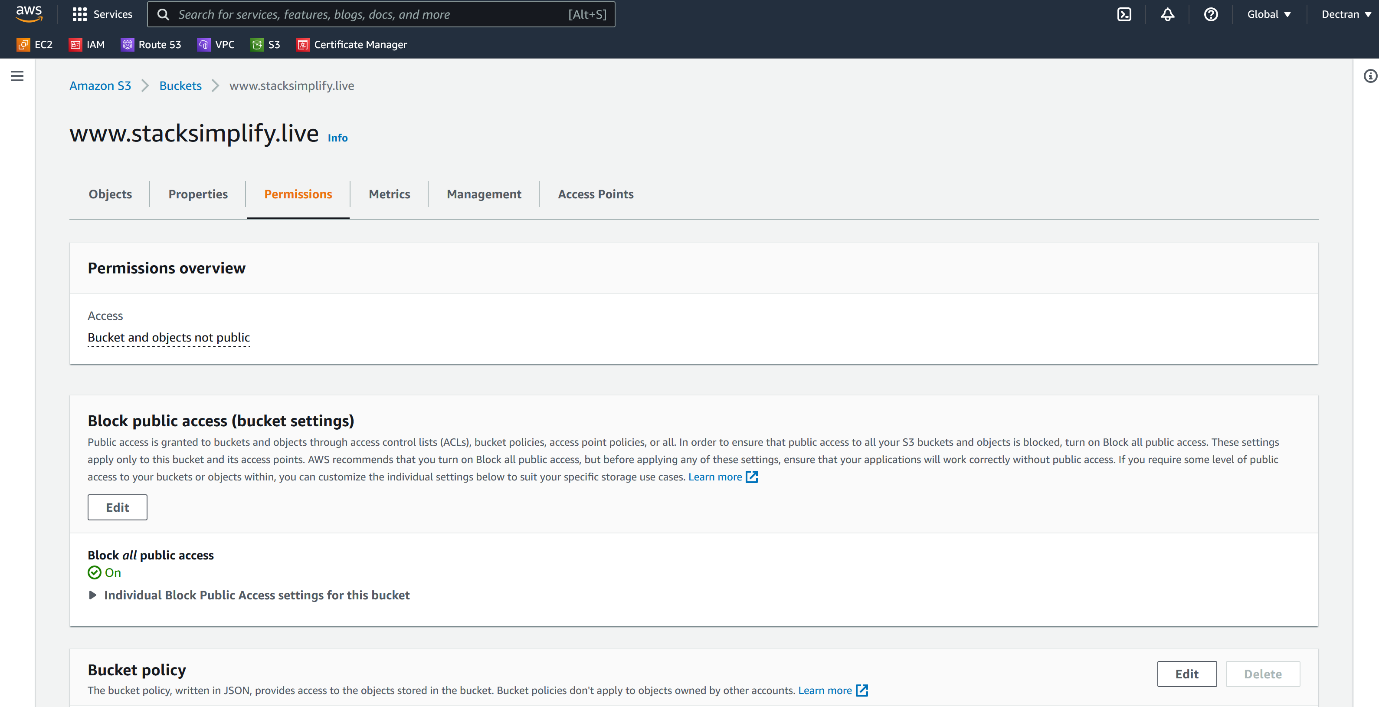


--- this is your website hosting and click on the url to open.

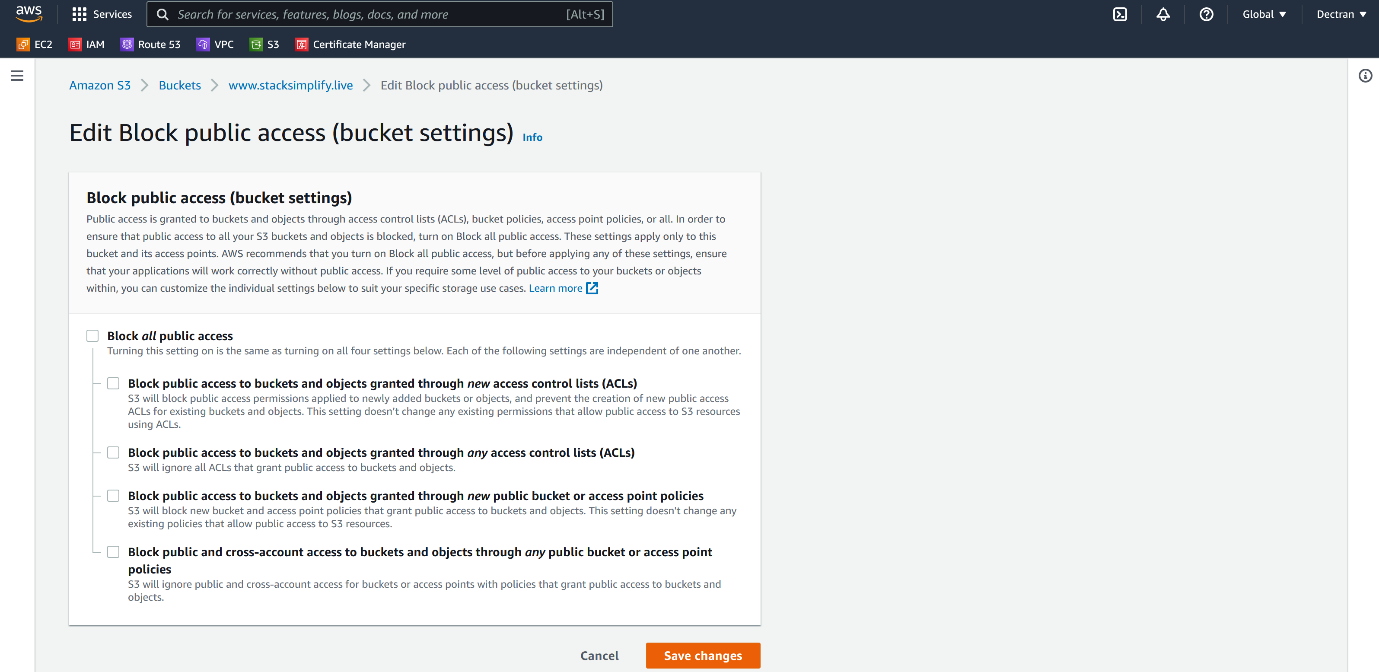


--- **note** – by default this url is not accessible on the internet. We need to enable the url publicly.

**Permissions**

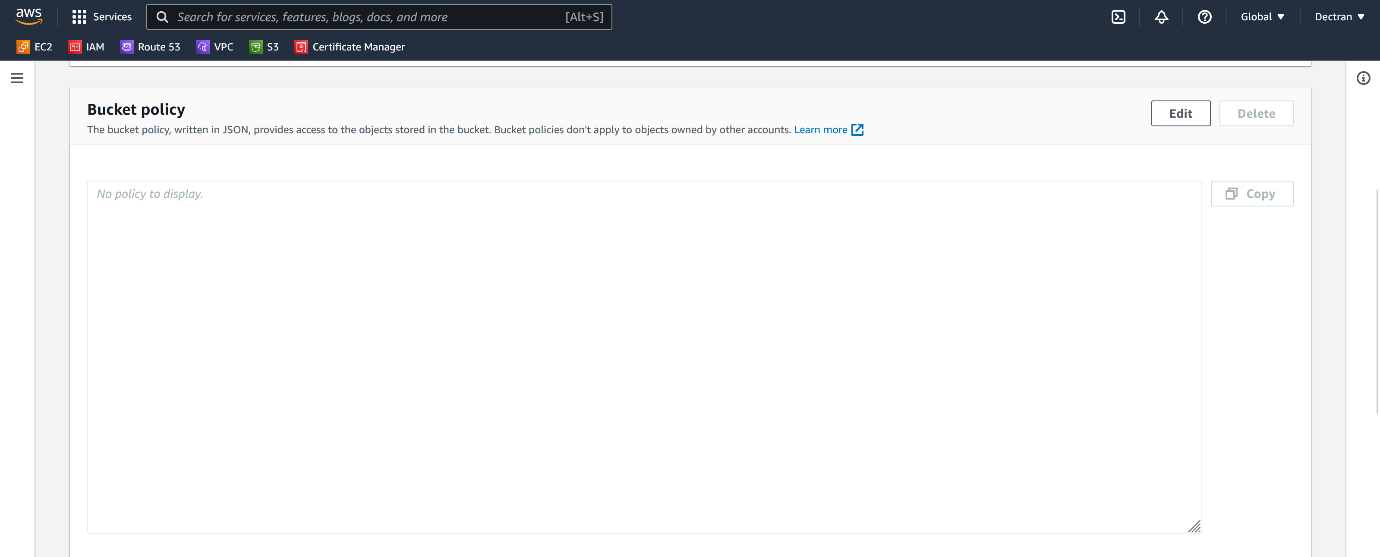


--- click on edit option.



--- click on save changes.

--- scroll down, you will see another option called bucket policy and click on edit option.



--- Click on edit option.

{

    "Version": "2012-10-17",

    "Statement": [

        {

            "Sid": "PublicReadGetObject",

            "Effect": "Allow",

            "Principal": "\*",

            "Action": "s3:GetObject",

            "Resource": "arn:aws:s3:::stacksimplify.live/\*"

        },

        {

            "Sid": "2",

            "Effect": "Allow",

            "Principal": {

                "AWS": "arn:aws:iam::cloudfront:user/CloudFront Origin Access Identity YOUR ID Access Identity E3M8L5UQ96KPB1"

            },

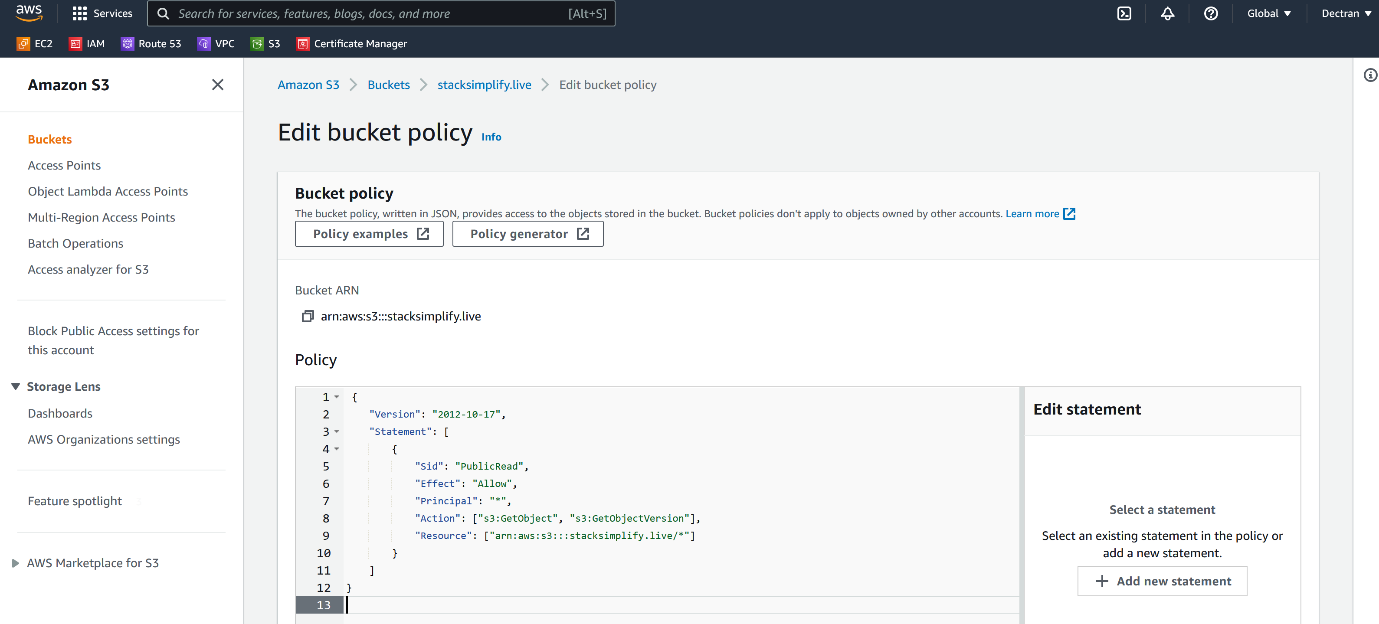
            "Action": "s3:GetObject",

            "Resource": "arn:aws:s3:::stacksimplify.live/\*"

        }

    ]

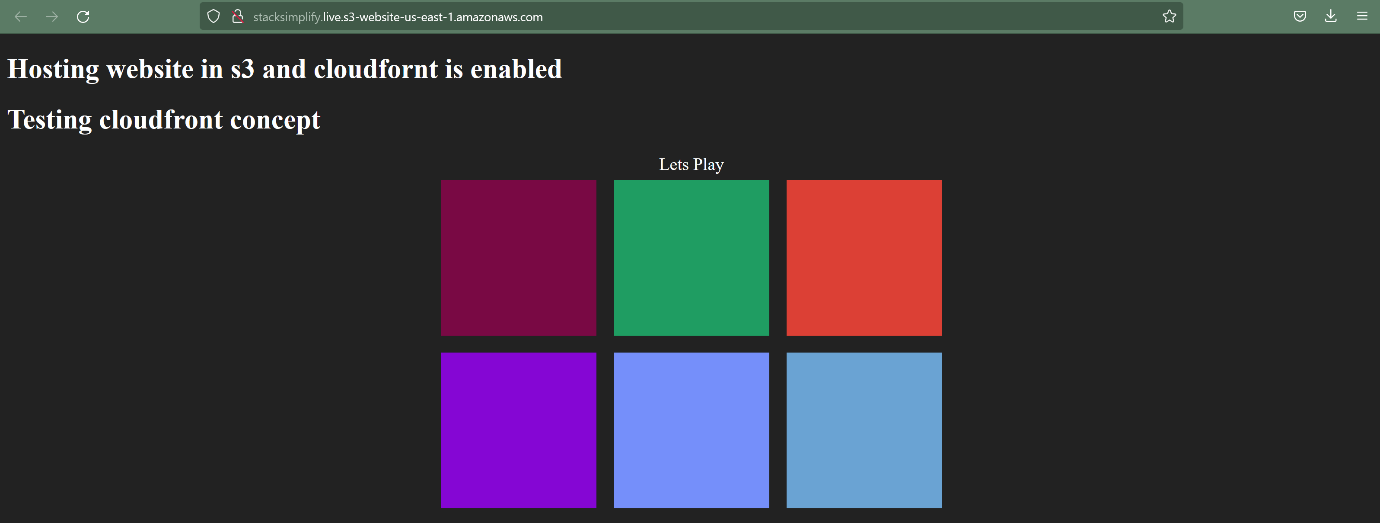
}



--- click on save changes.

**Website testing**

--- <http://stacksimplify.live.s3-website-us-east-1.amazonaws.com> – hit the url on cloud to see whether it is accessible.

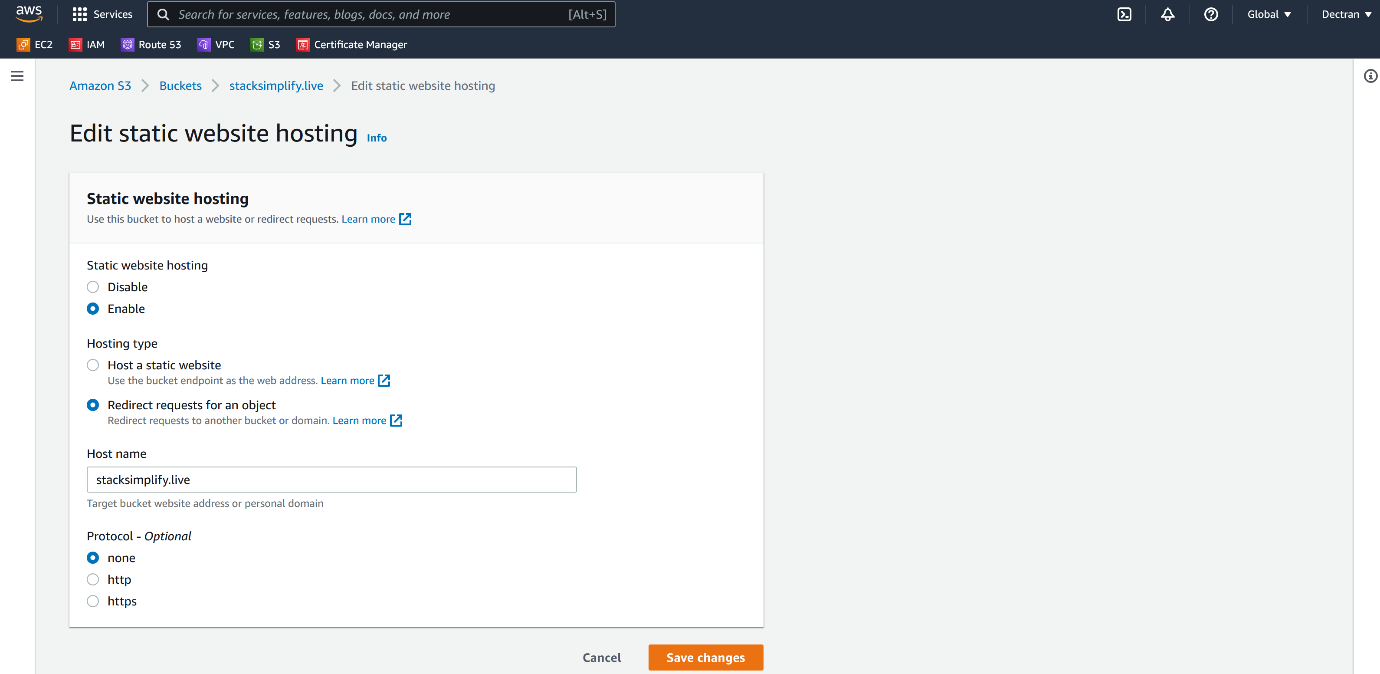


--- note – our website working.

**Redirect website**

--- if someone access url like [www.stacksimplify.live](http://www.stacksimplify.live) then it should be routed to <https://stacksimplify.live>.

--- for this go to the second bucket [www.stacksimplify.live](http://www.stacksimplify.live), go to properties and click on static website hosting.



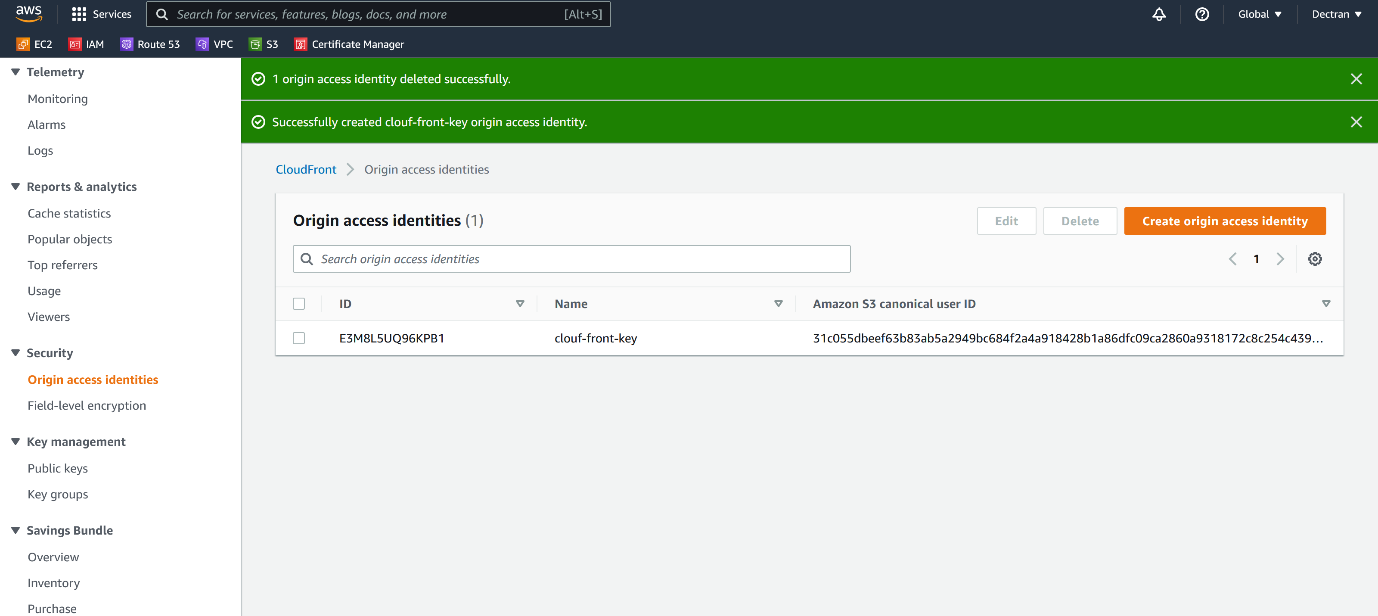
--- give the bucket name that you want to redirect your request. click on save changes.

**Permissions**

--- give public access.

**Cloudfront**

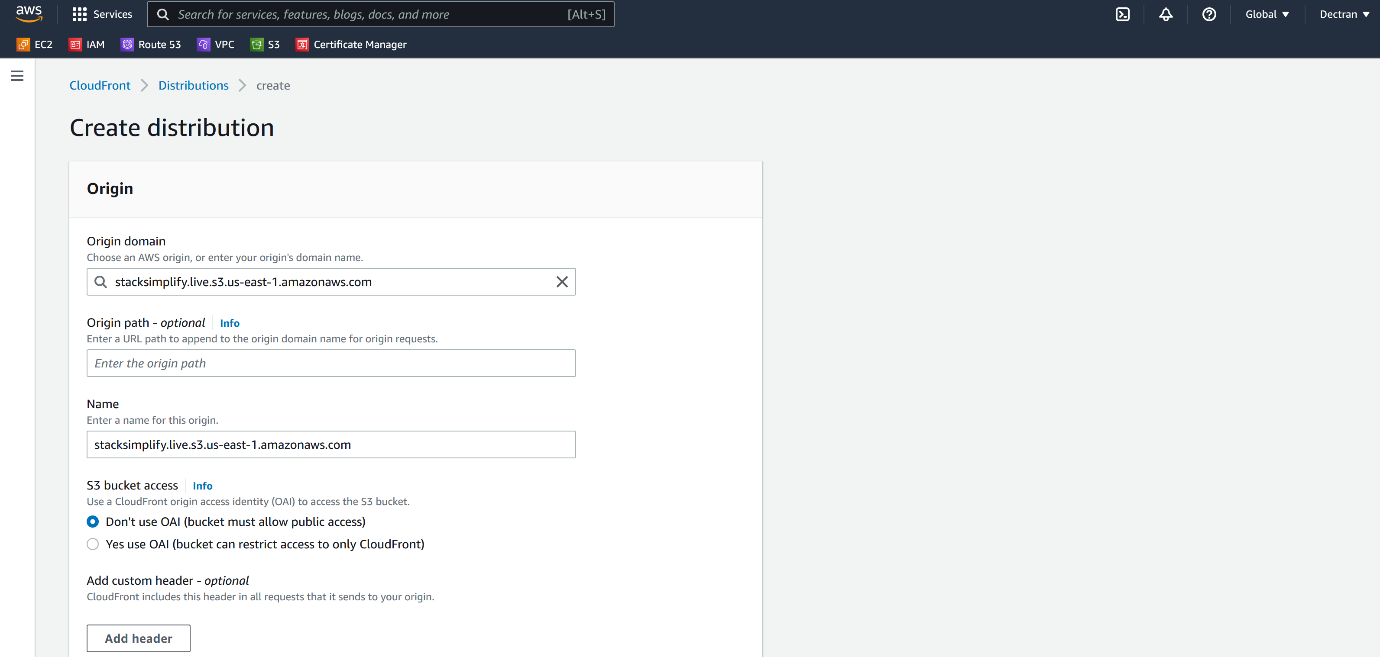
--- create an origin access identity, which is present in security.

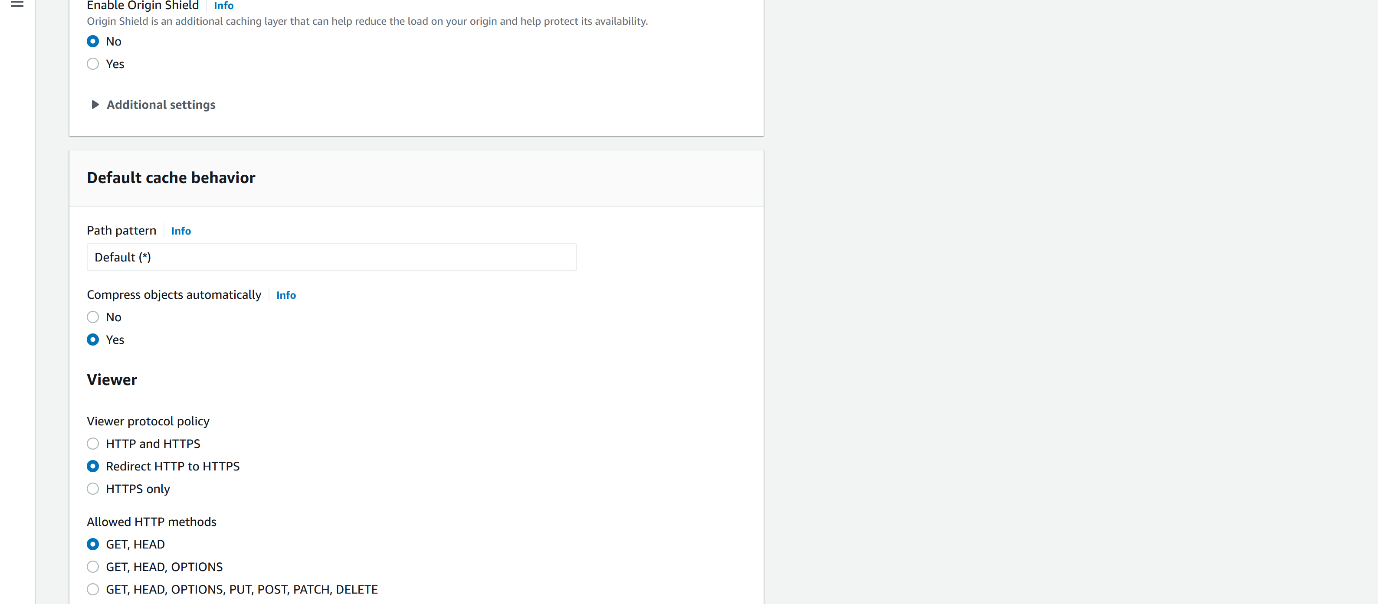


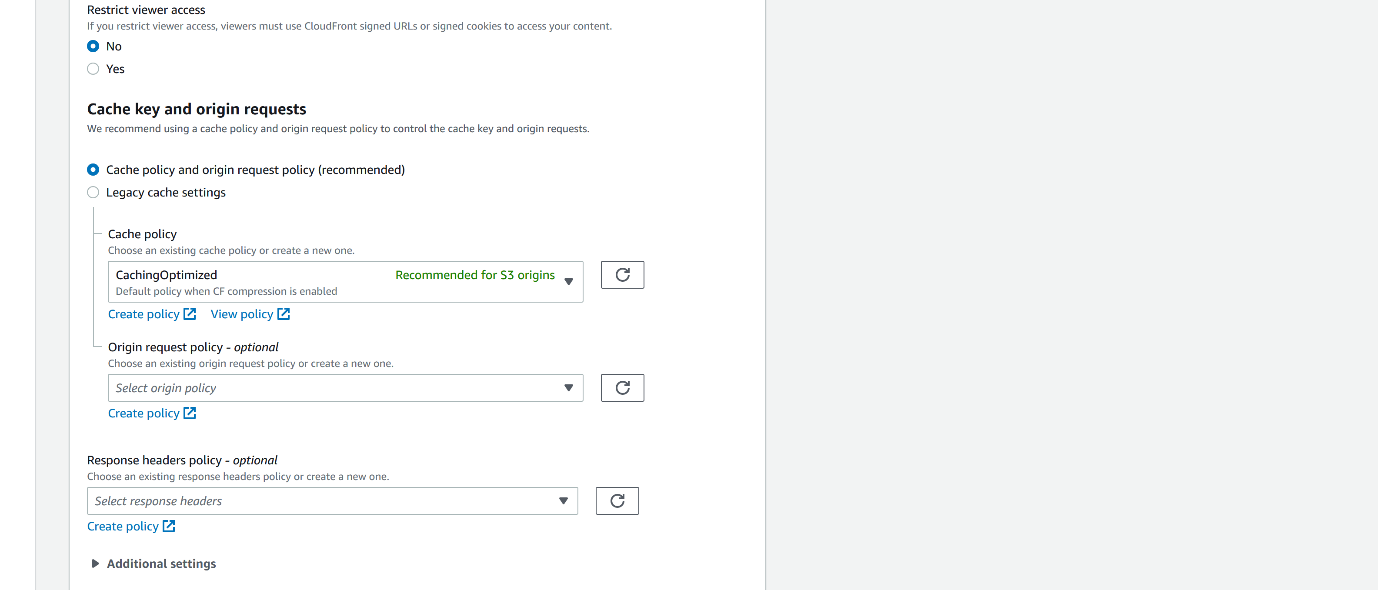
--- note – copy the id and use this s3 bucket policy.

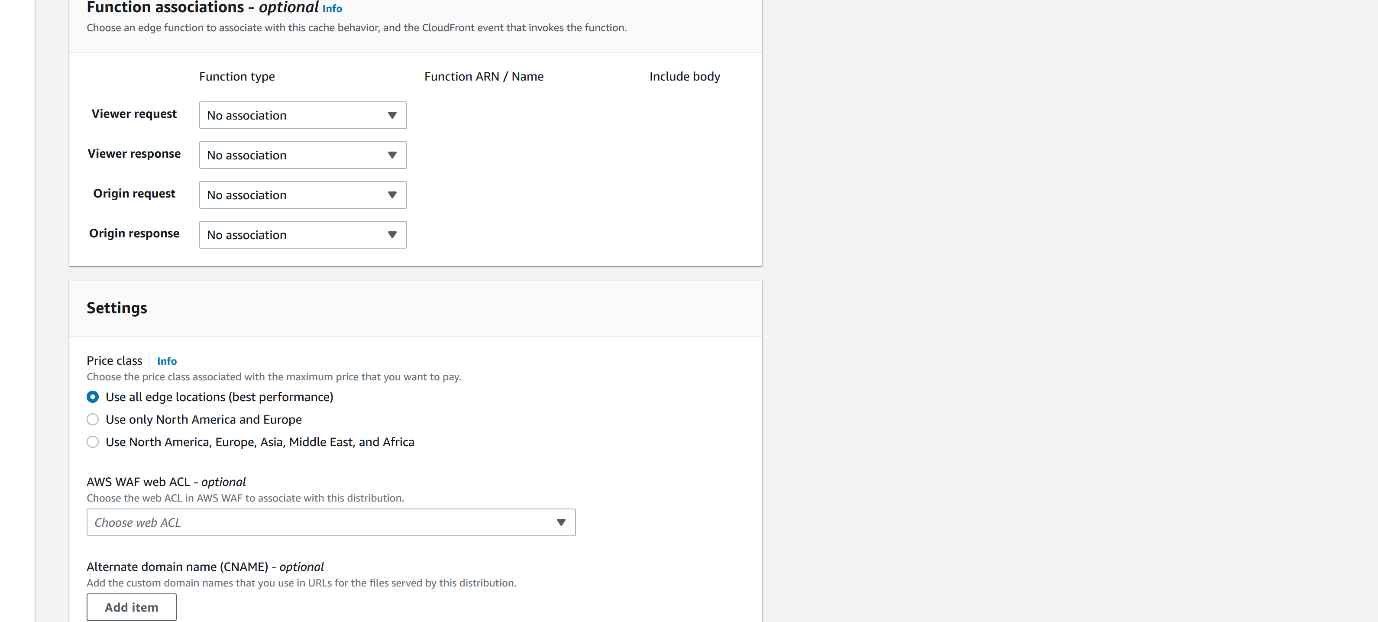
--- now, I place cloudfront Infront of the s3 bucket.

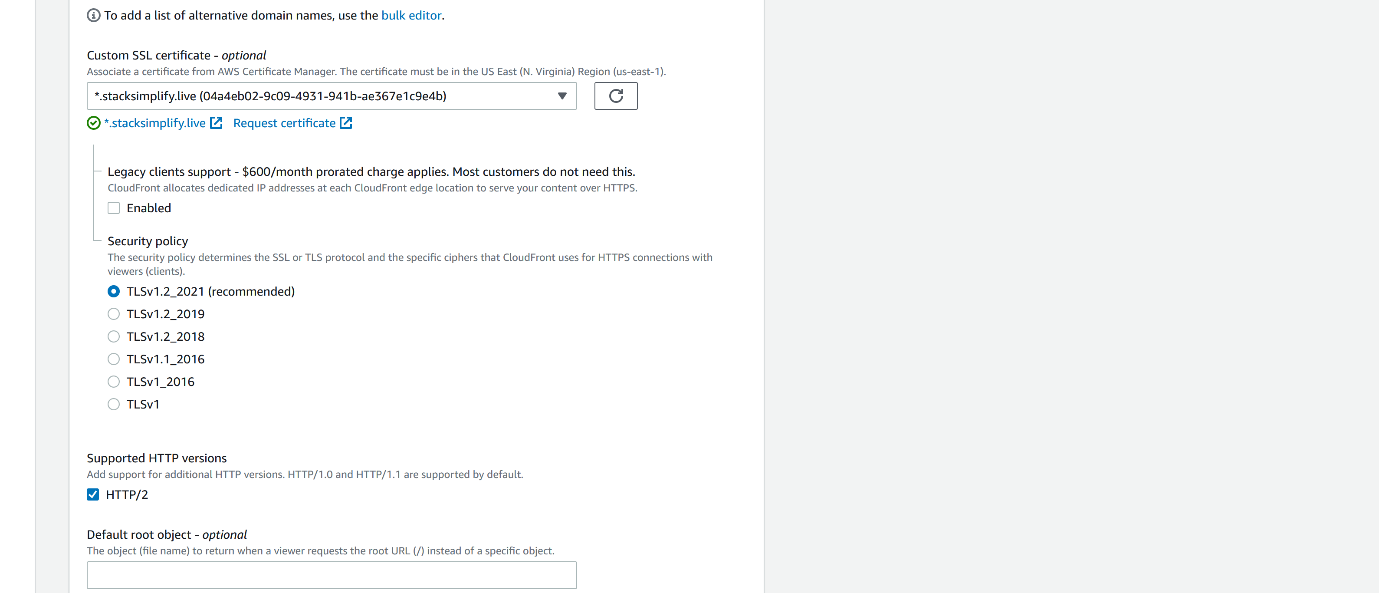
--- go to cloudfront and click on distribution.

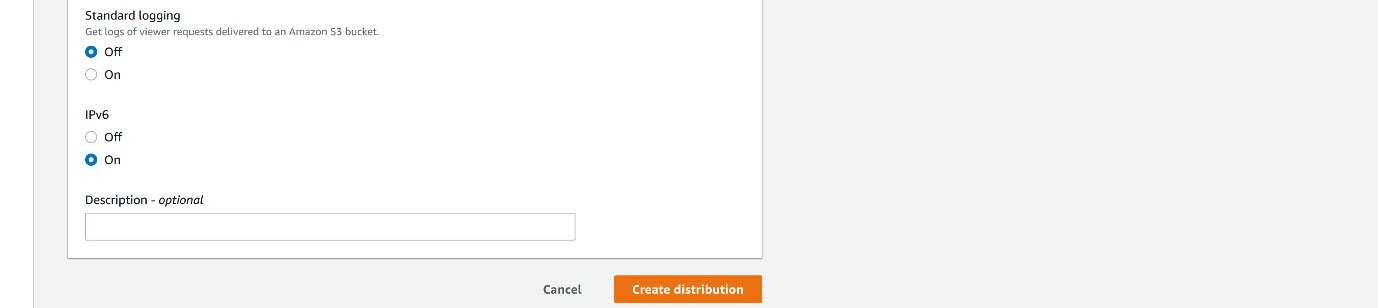




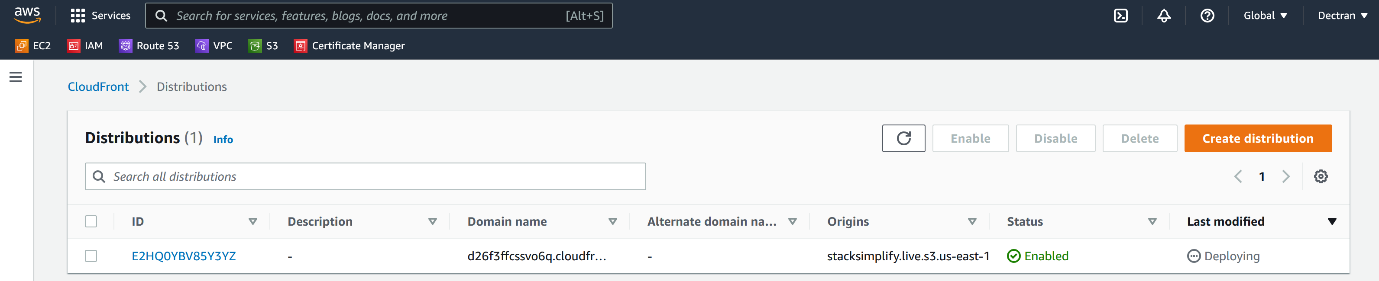




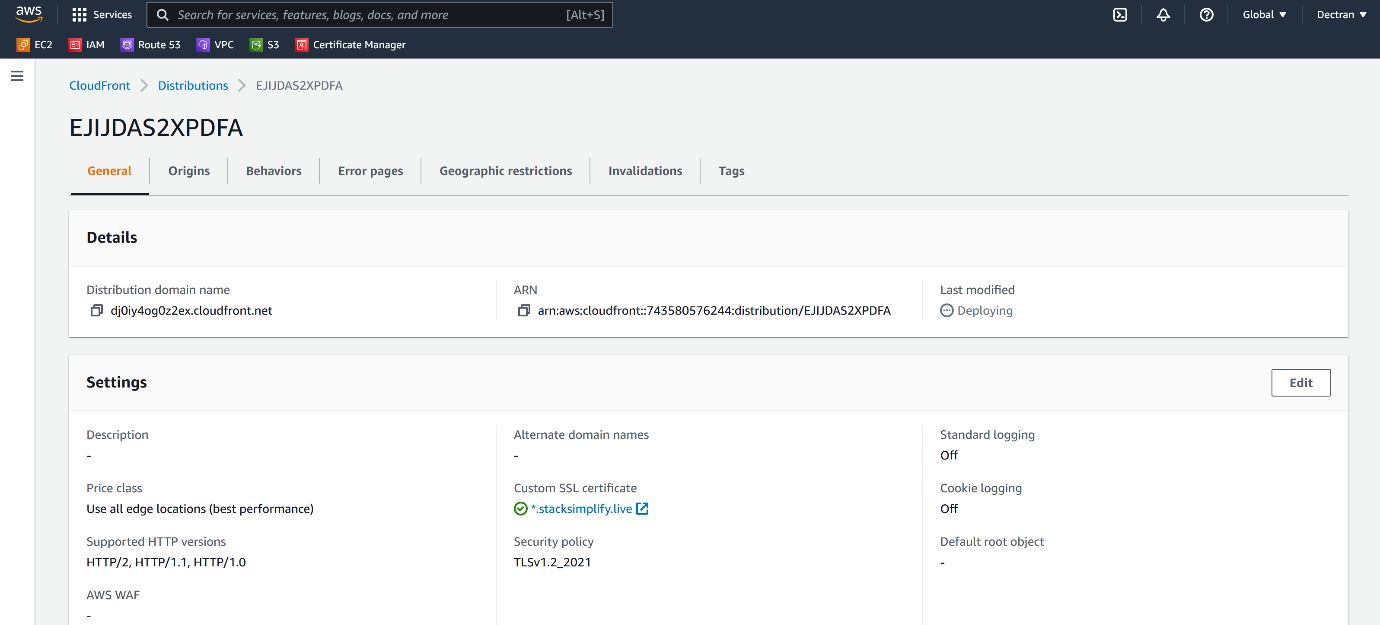




--- click on save changes.

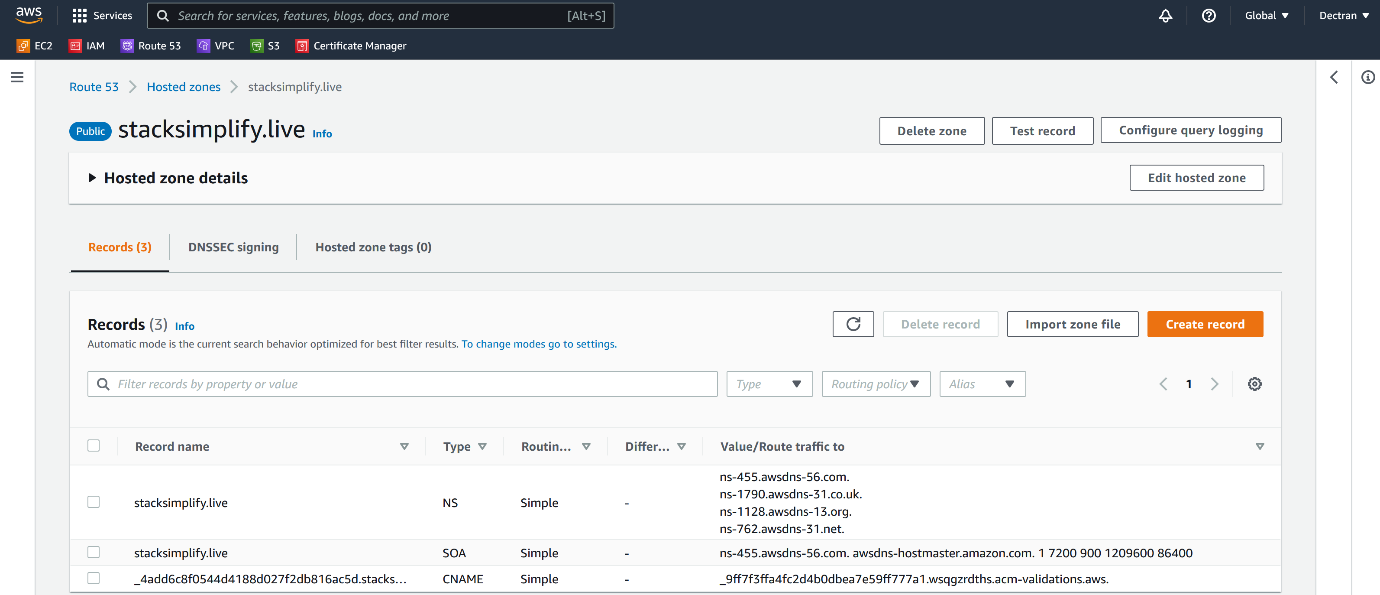


--- cloudfront distribution got created. Now clikc on the distribution.

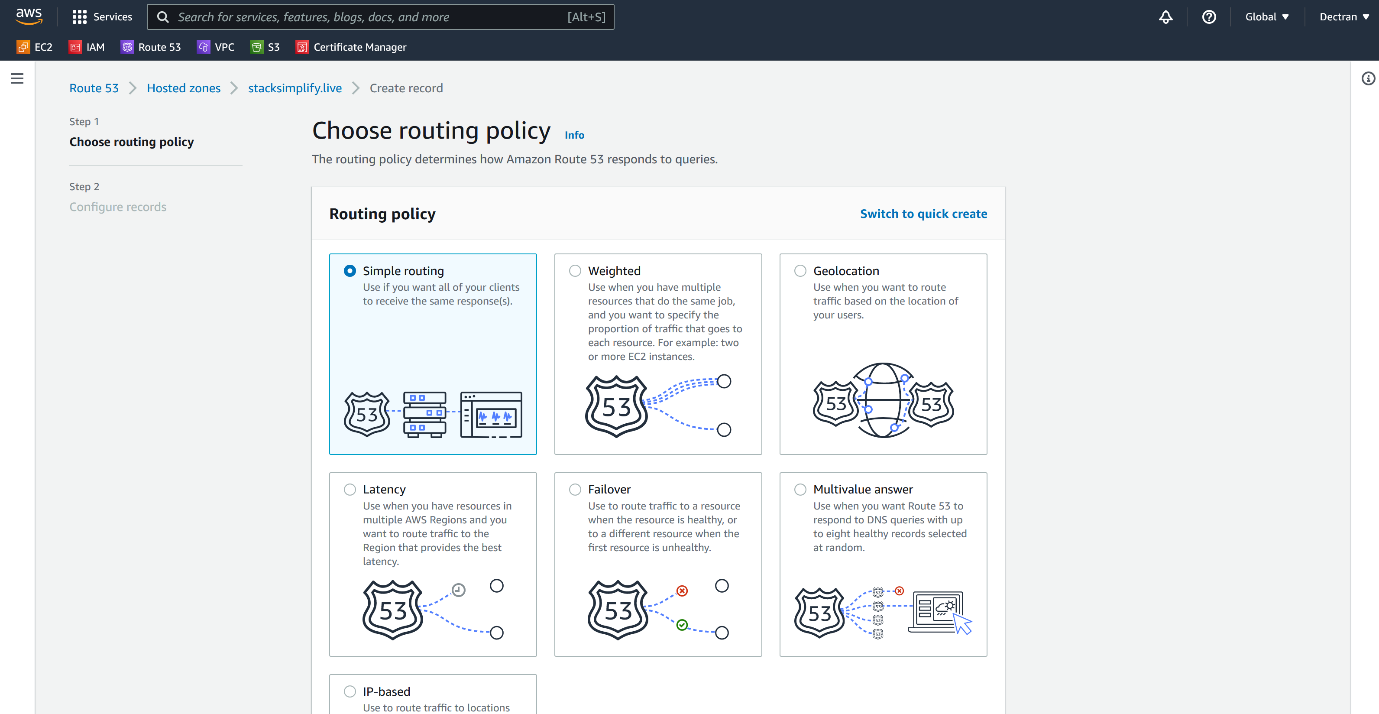


--- copy the distribution domain name and hit the domain name on separate tab.

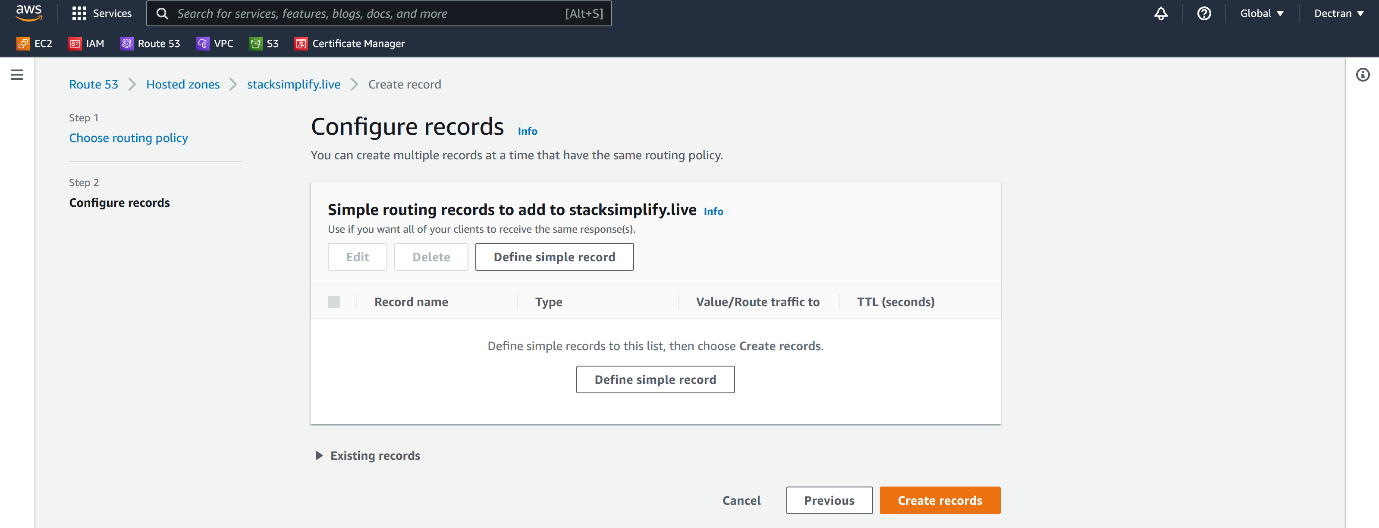
**Route53**

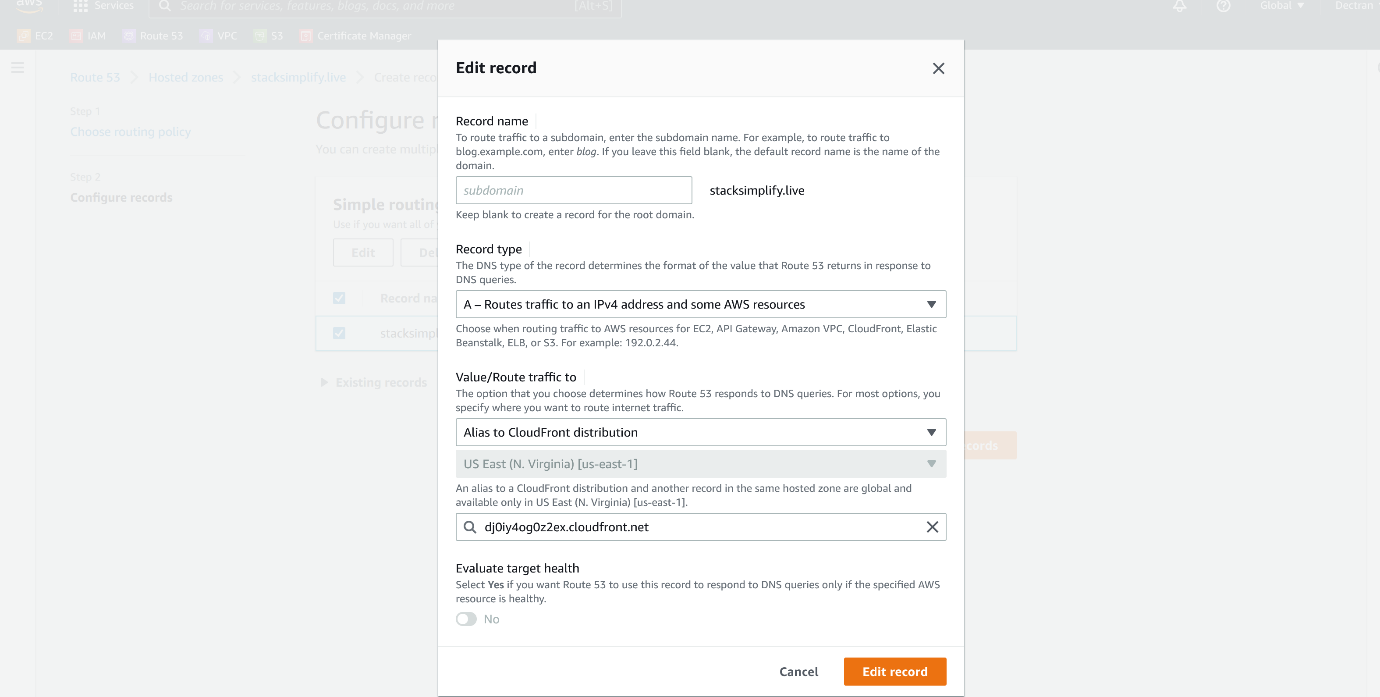


--- click on add a record.



--- click on next.

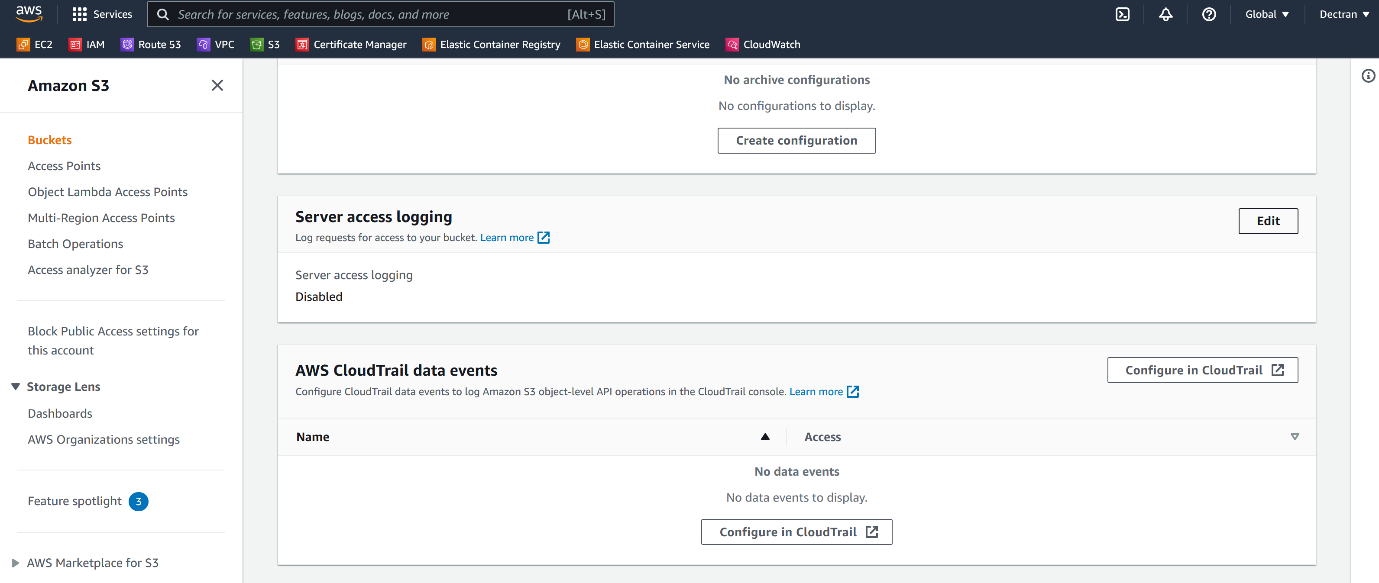




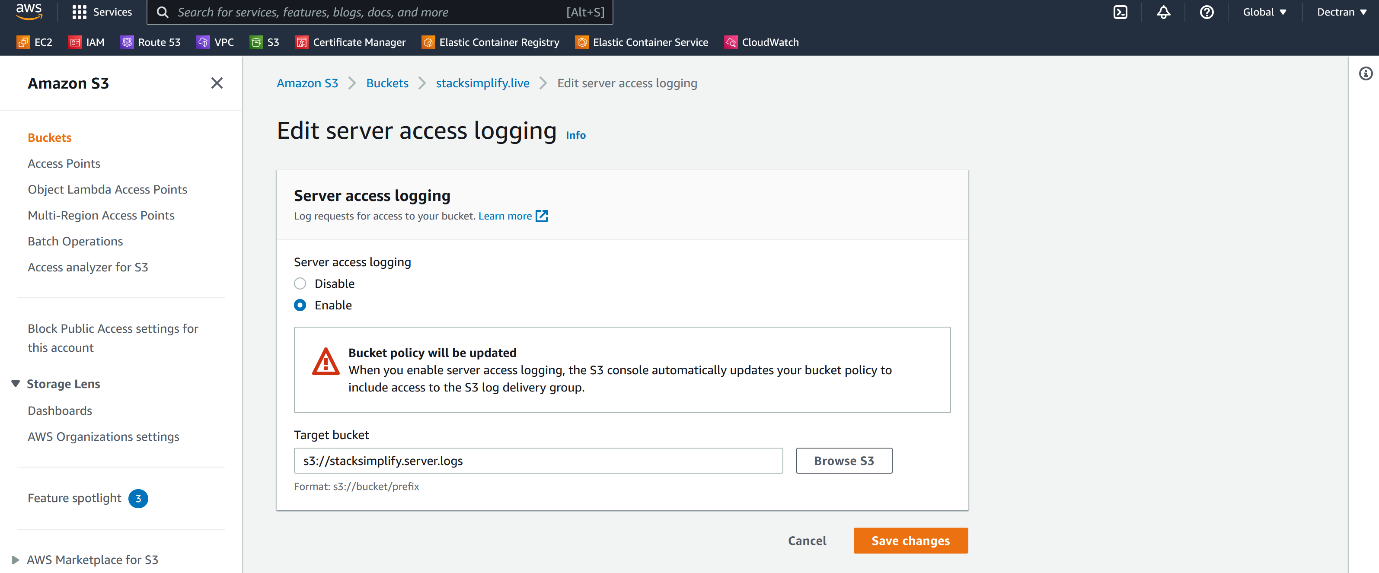
--- click on define record.

**Enabling logs for our website**

--- under properties, click on server access logging.

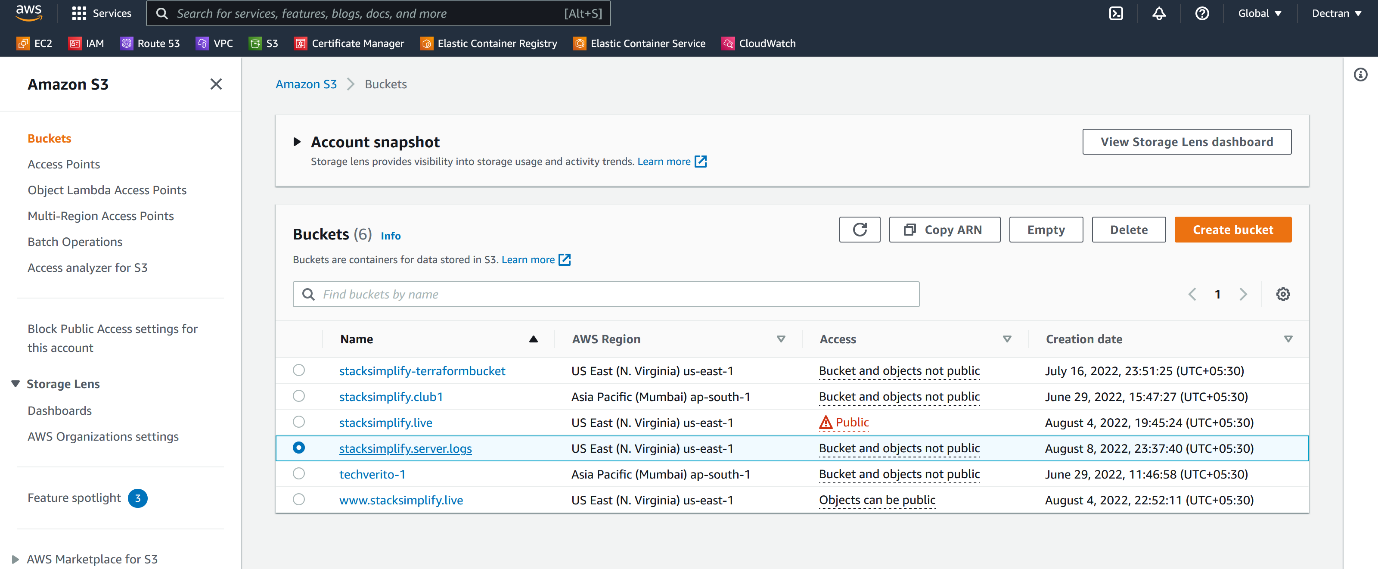


--- click on server access logging.



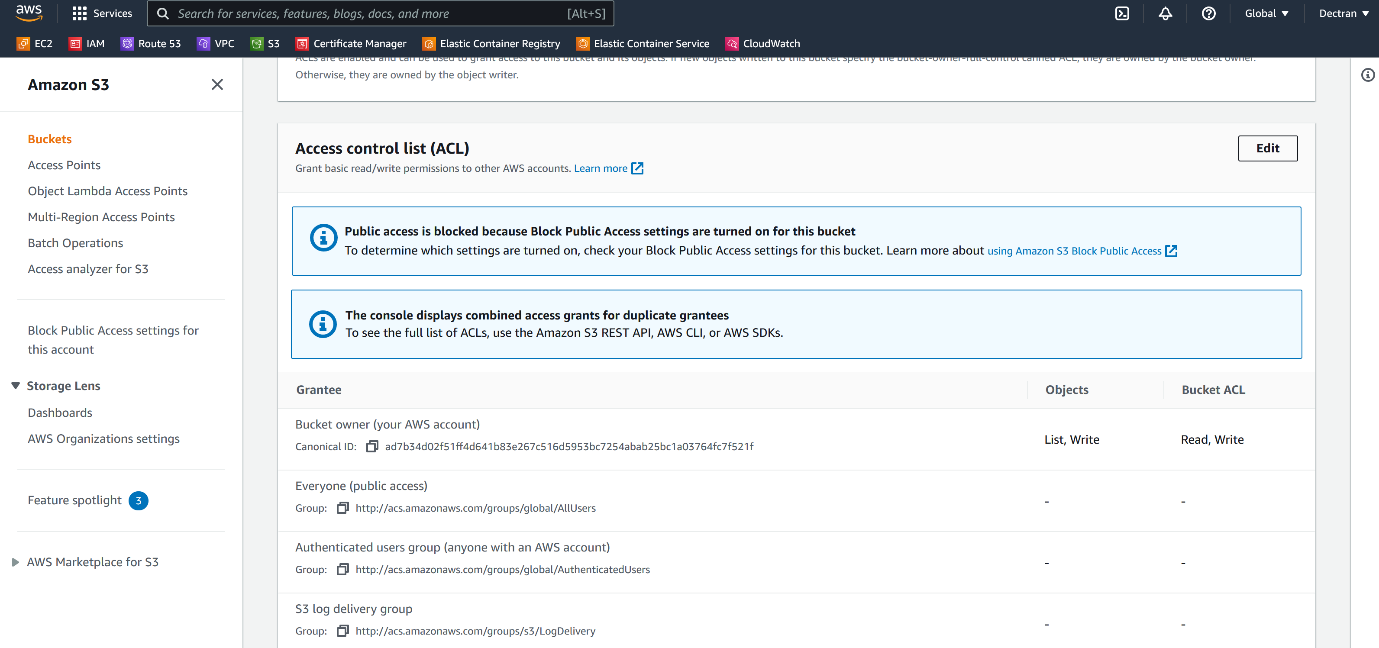
--- I have created a separate bucket for logging and select the bucket here.

**Enable s3 log deleivery option in webserver loging bucket (stacksimplify.server.logs)**

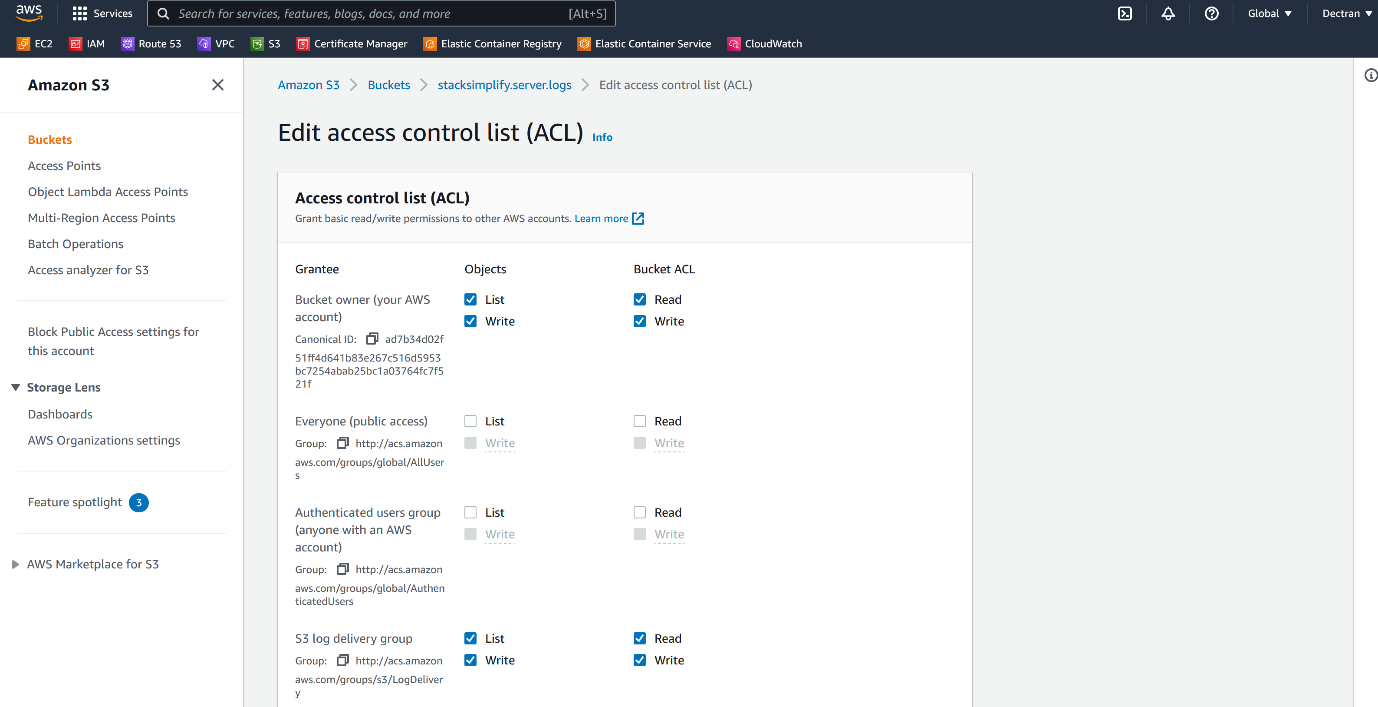


--- click on the stacksimplify.server.logs

--- under permission, you will find NAC and edit NACL permissions.



--- click on edit option.

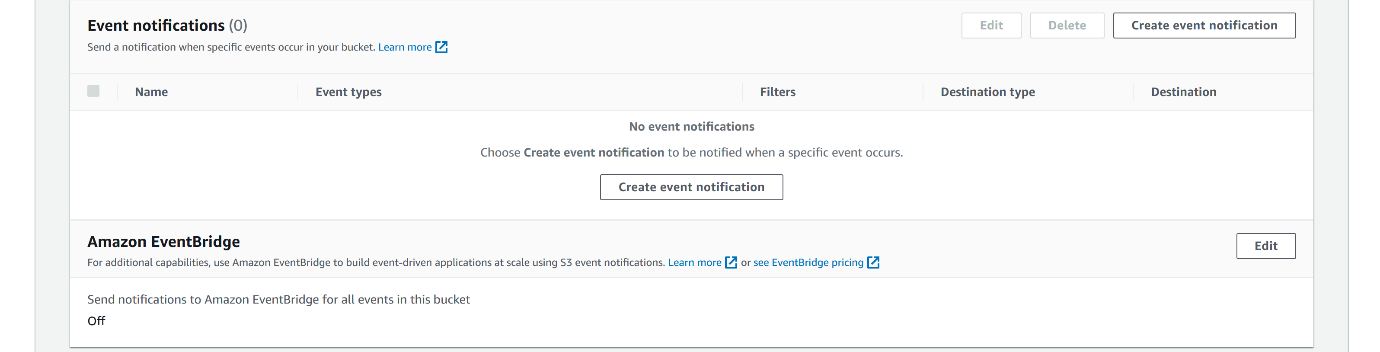


--- check all the boxes which are present under s3 log delivery group.

--- **note** – now this bucket will allow logs form our website.

**Event notifications**

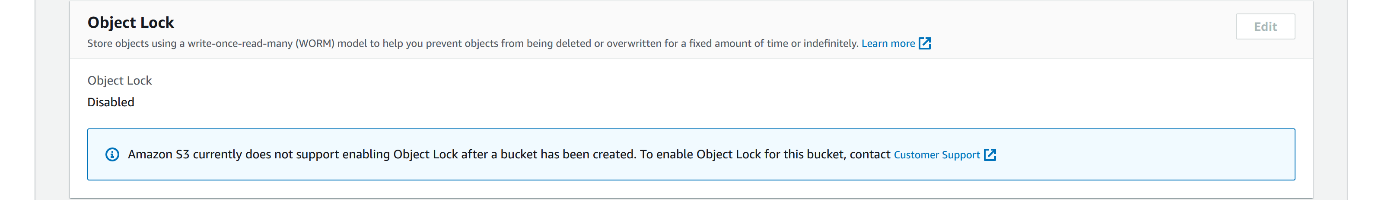
--- it is present under properties.



--- **note** – you want to get notification when somebody uploading files to bucket.

**Object lock**

--- it is present under properties lock.



--- **note** – you want to lock objects, it will not allow delete those objects form the s3 bucket.

--- note – we need to allow this option from the starting.