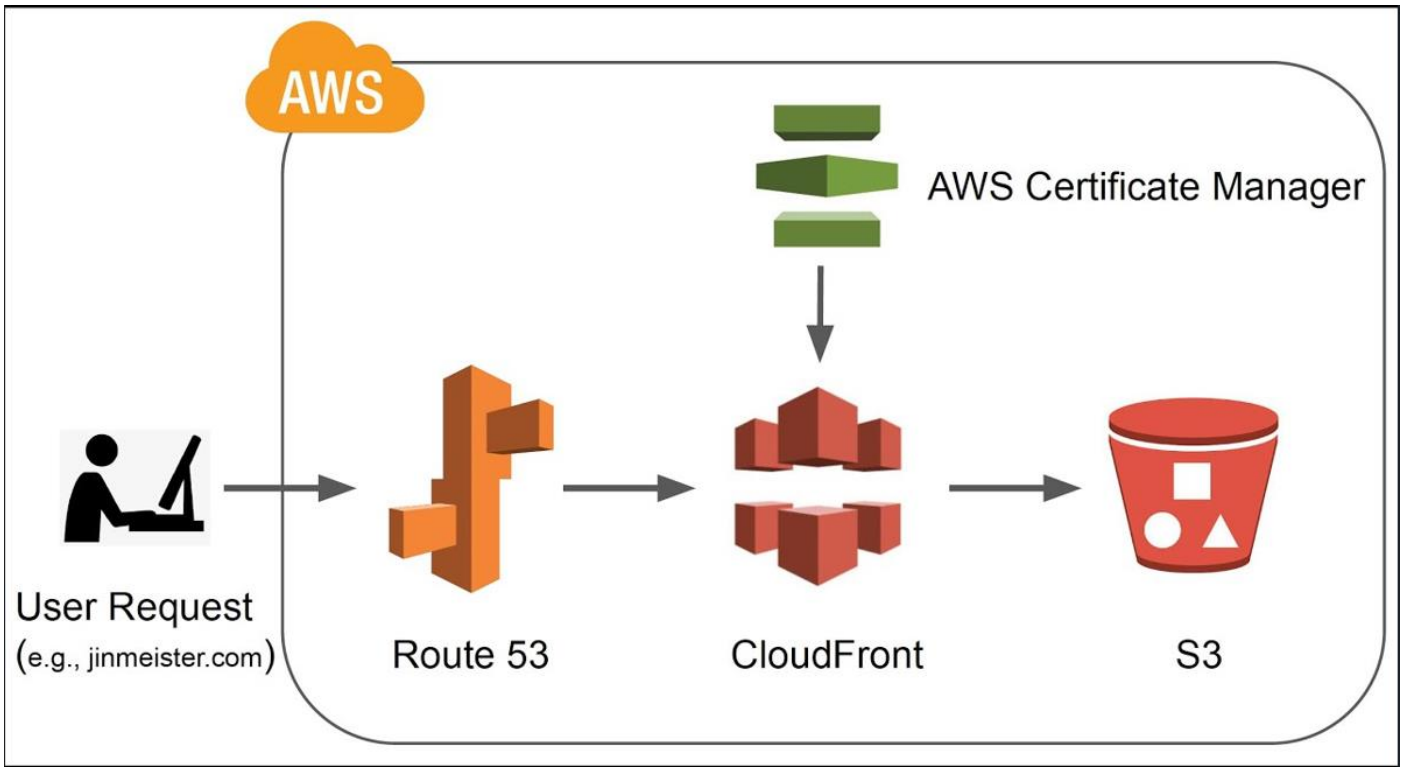
Hosting Static Website on AWS S3



Services That we are using to host the static website:

1. **Route 53**
2. **Cloud Front**
3. **S3(simple Storage Service)**
4. **AWS Certificate Manager)**

Lets understand Services little bit that we are going to use it.

1.Route 53: Amazon Route 53 is a highly available and scalable cloud Domain Name System (DNS) web service.
It is basically designed for developers and corporate to route the end users to Internet applications by translating human-readable names like www.sanoj.homes into the numeric IP addresses like 192.168.1.1 that computers use to connect to each other.

You cannot use Amazon Route 53 to connect your on-premises network with AWS Cloud.

2.Cloudfront: It is a content delivery network service that speeds up the distribution of static(Mostly S3) and dynamic web content (EC2 for static/dynamic) to the users.

It keeps the content on the edge locations so that users can retrieve it easily whenever he requests it.
It delivers the content with the best possible performance by routing the user at the closest edge location.

3.S3(Simple Storage Service): S3, is the object storage service provided by AWS. It is probably the most commonly used, go-to storage service for AWS users given the features like extremely high availability, security, and simple connection to other AWS Services.

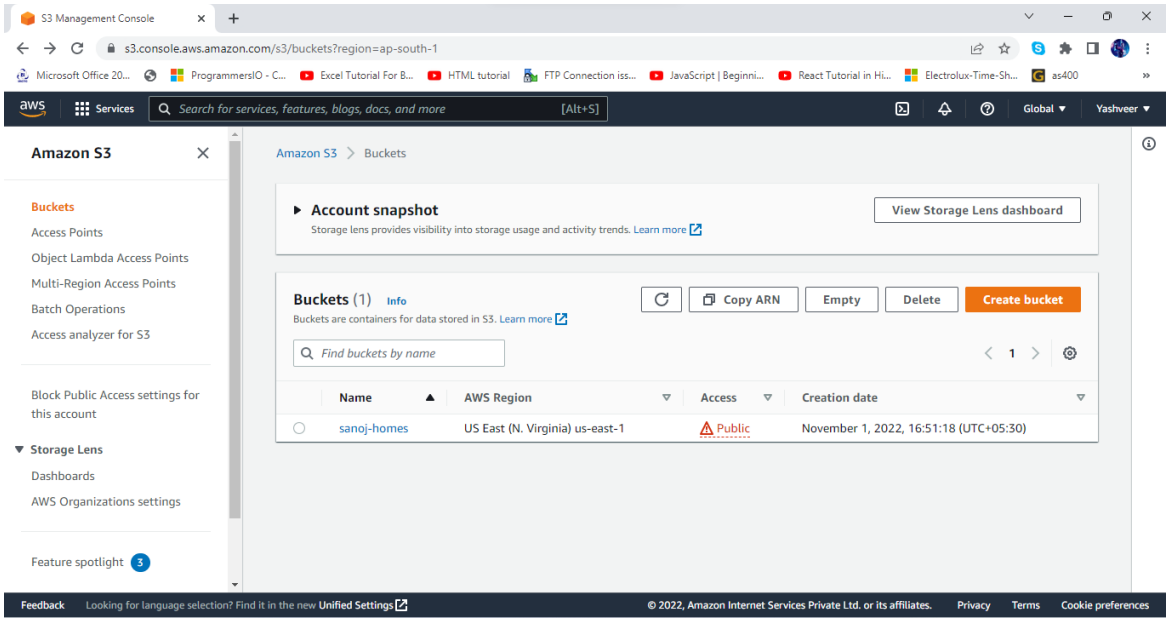
AWS S3 can be used by people with all kinds of use cases like mobile/web applications, big data, machine learning and many more.

4.AWS Certificate Manager: is designed to simplify and automate many of the tasks traditionally associated with provisioning and managing SSL/TLS certificates. ACM takes care of the complexity surrounding the provisioning, deployment, and renewal of digital certificates for no extra cost.

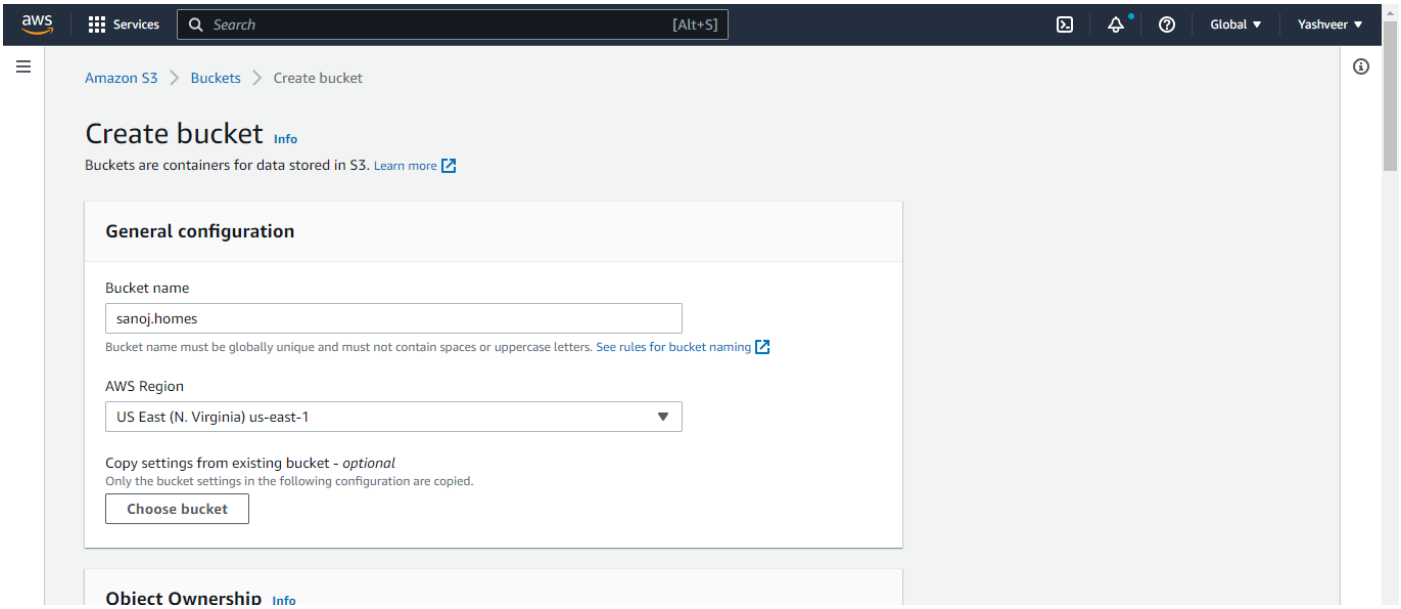
Now we are good to go to host the static website on S3 using route 53, S3, ACM, CloudFront.

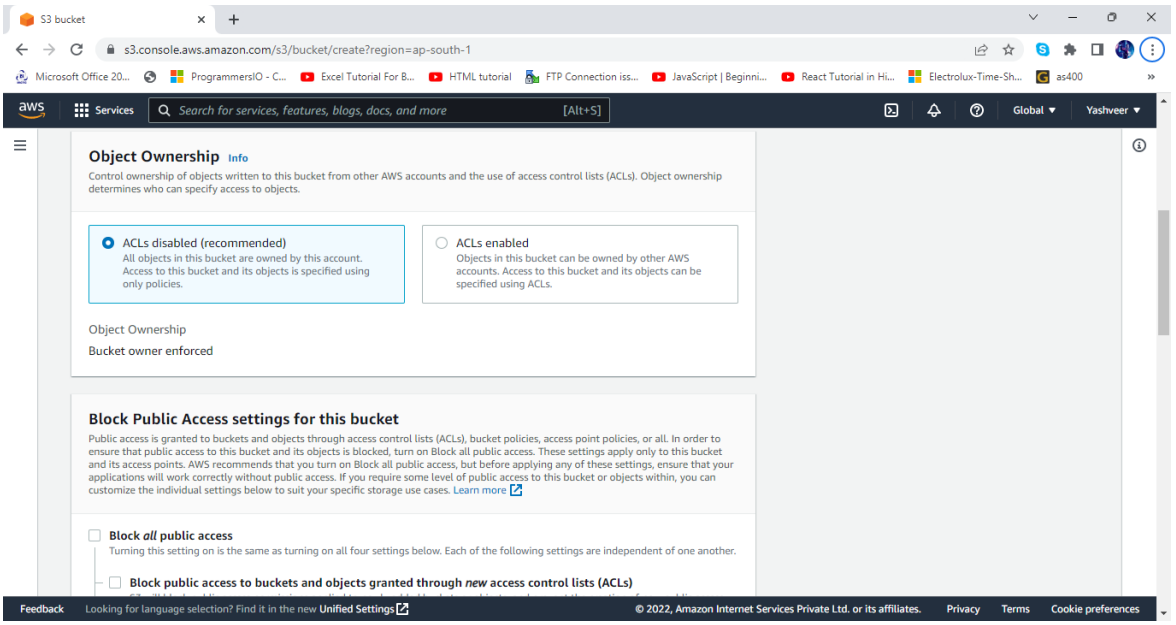
Setting Up S3:

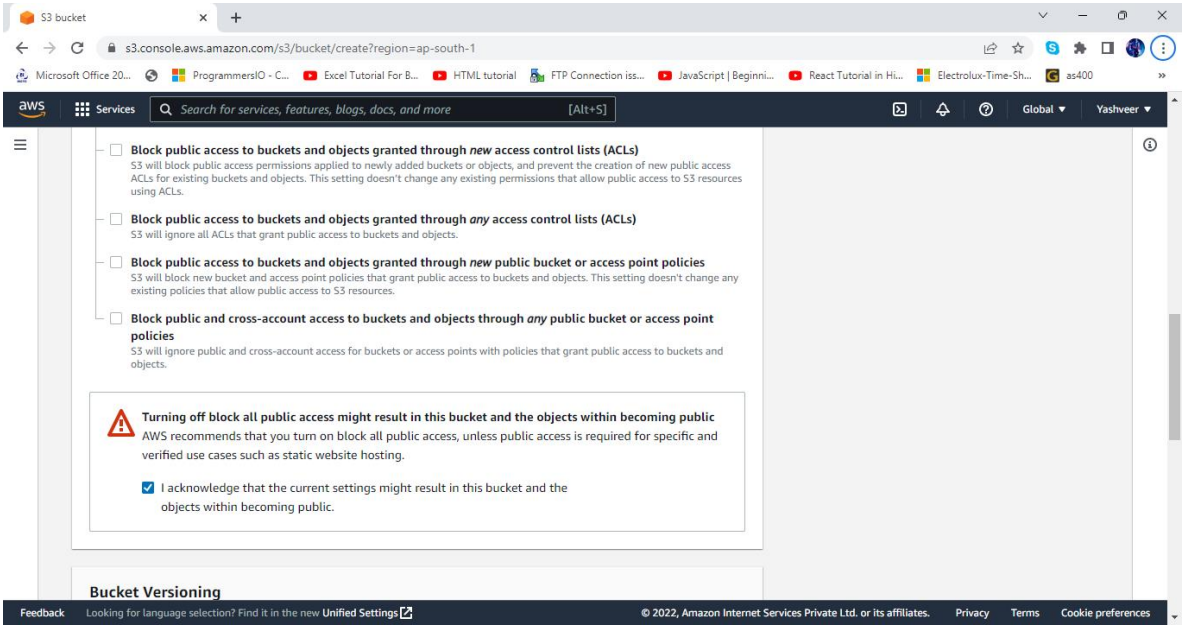
Step1: Search the S3 from AWS console you will see below kind interface.

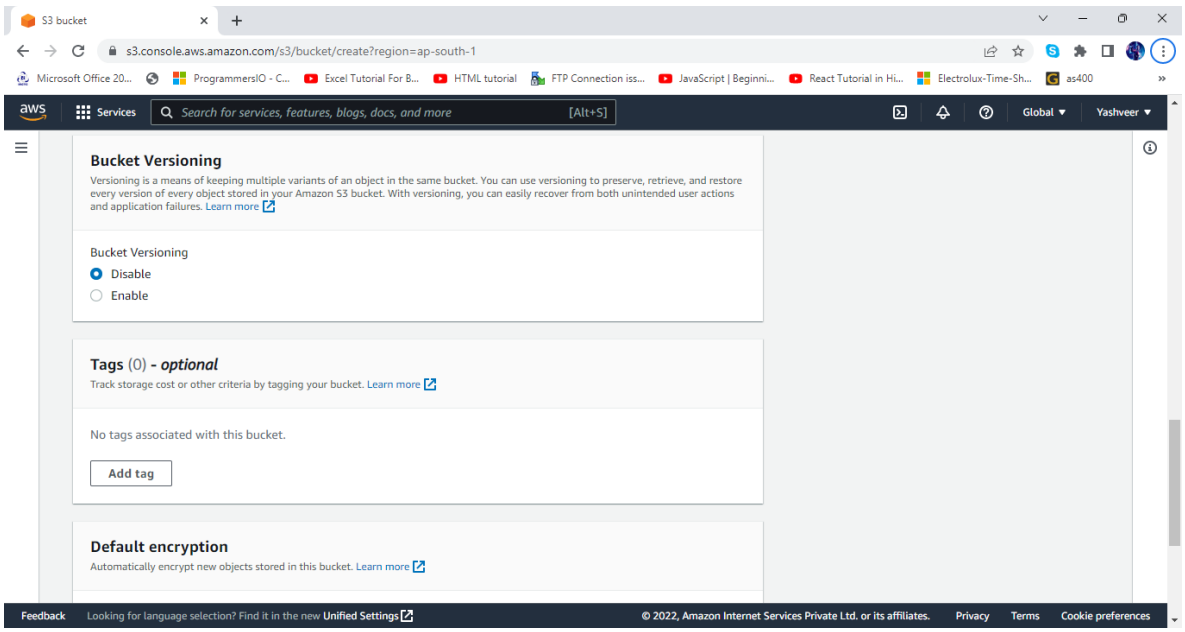


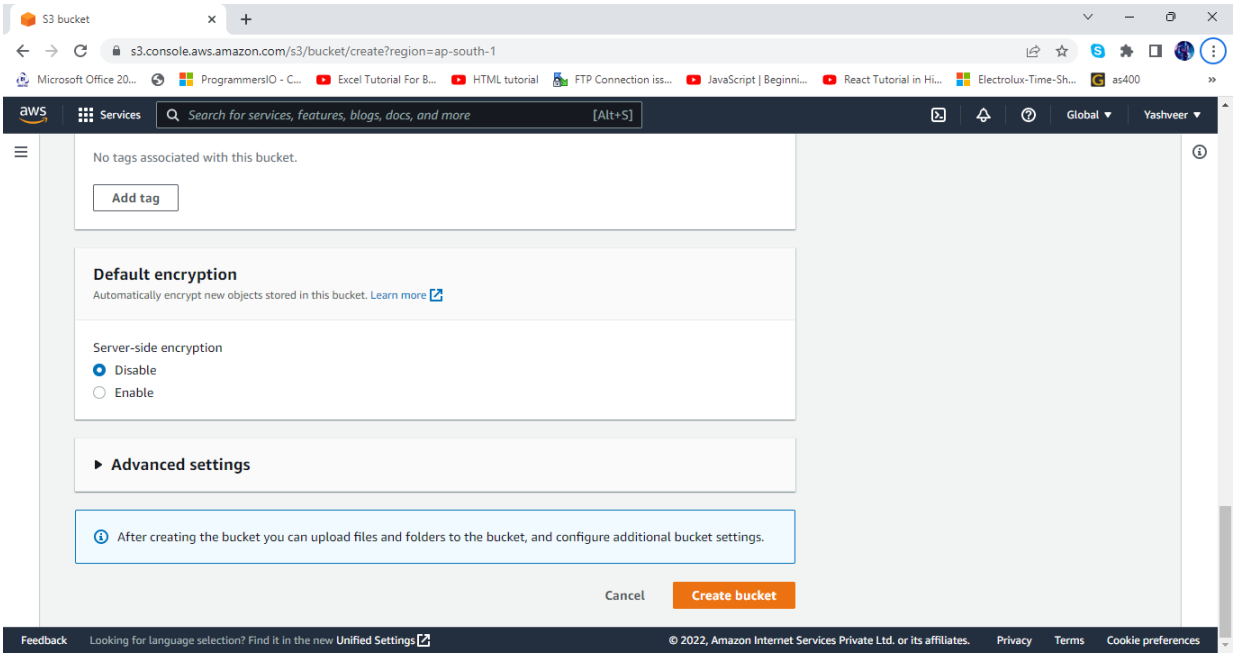
Step2: Click on the Create Bucket, once you click on the create bucket you will see below interface.



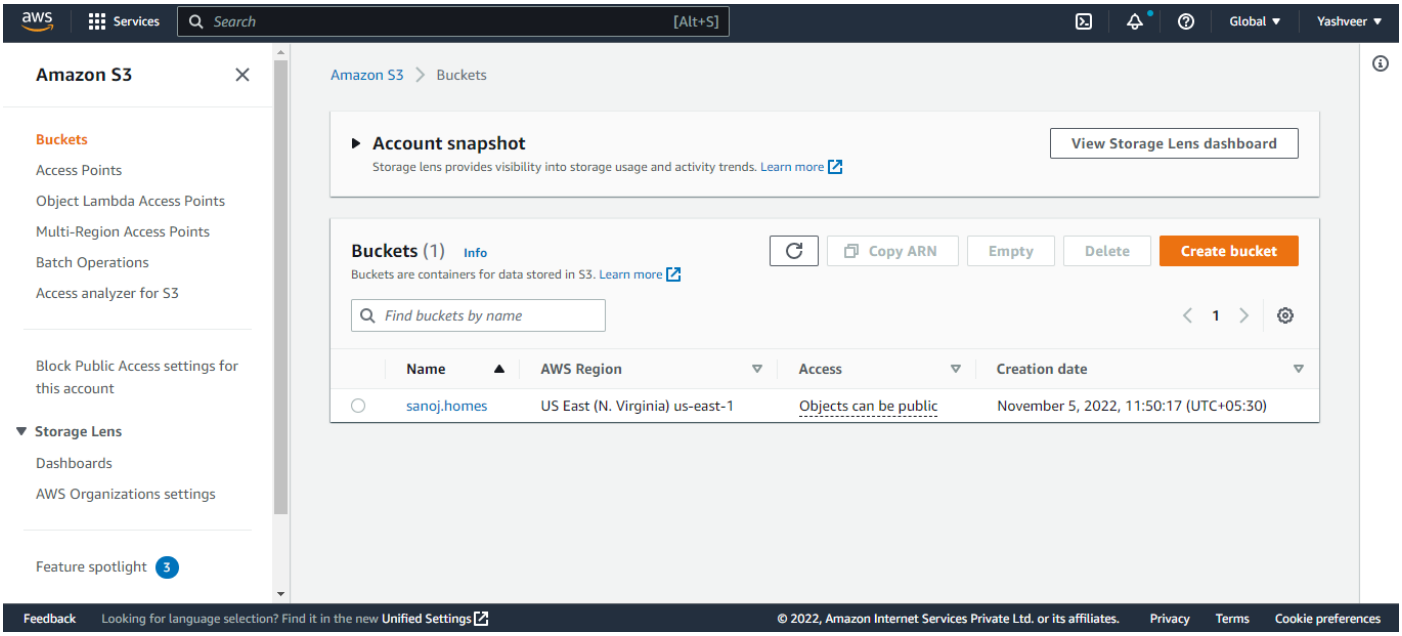

S3 bucket
s3.console.aws.amazon.com/s3/bucket/create?region=ap-south-1
Services
Search for services, features, blogs, docs, and more
[Alt+S]
Global
Yashveer
Object Ownership Info
Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.
ACLs disabled (recommended)
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.
ACLs enabled
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.
Object Ownership
Bucket owner enforced
Block Public Access settings for this bucket
Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more
Block all public access
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.
Block public access to buckets and objects granted through new access control lists (ACLs)
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Internet Services Private Ltd. or its affiliates.
Privacy
Terms
Cookie preferences


S3 bucket
s3.console.aws.amazon.com/s3/bucket/create?region=ap-south-1
Services
Search for services, features, blogs, docs, and more
[Alt+S]
Global
Yashveer
Block public access to buckets and objects granted through new access control lists (ACLs)
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
Block public access to buckets and objects granted through any access control lists (ACLs)
S3 will ignore all ACLs that grant public access to buckets and objects.
Block public access to buckets and objects granted through new public bucket or access point policies
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
Block public and cross-account access to buckets and objects through any public bucket or access point policies
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.
Turning off block all public access might result in this bucket and the objects within becoming public
AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.
I acknowledge that the current settings might result in this bucket and the objects within becoming public.
Bucket Versioning
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Internet Services Private Ltd. or its affiliates.
Privacy
Terms
Cookie preferences

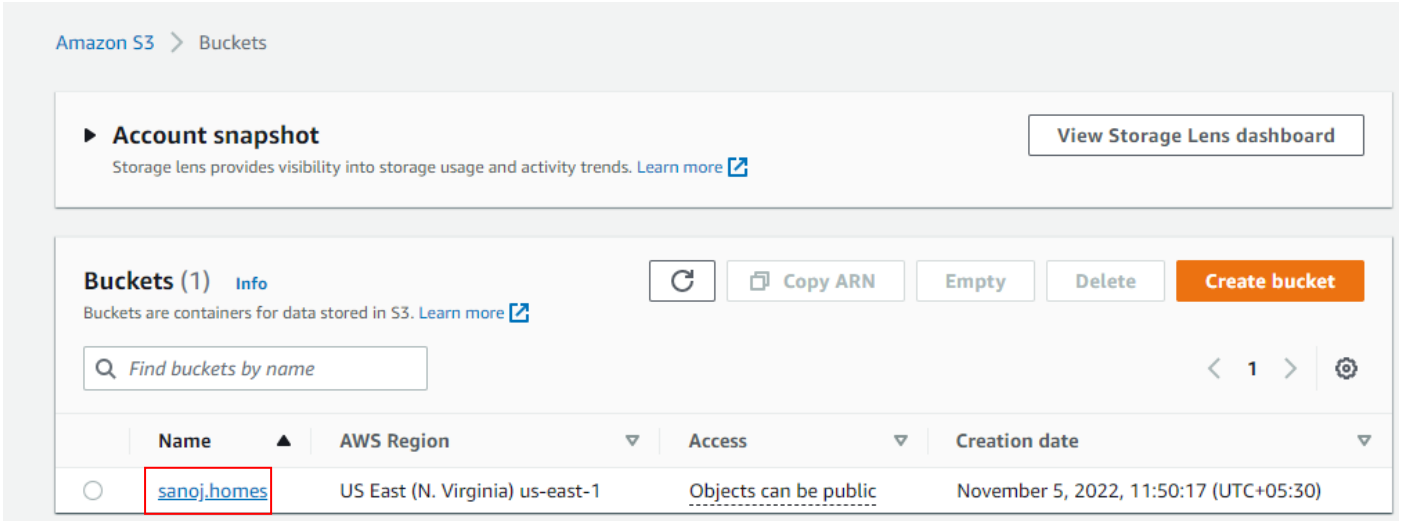

S3 bucket
s3.console.aws.amazon.com/s3/bucket/create?region=ap-south-1
Services
Search for services, features, blogs, docs, and more
[Alt+S]
Global
Yashveer
Bucket Versioning
Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. Learn more
Bucket Versioning
Disable
Enable
Tags (0) - optional
Track storage cost or other criteria by tagging your bucket. Learn more
No tags associated with this bucket.
Add tag
Default encryption
Automatically encrypt new objects stored in this bucket. Learn more
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Internet Services Private Ltd. or its affiliates.
Privacy
Terms
Cookie preferences



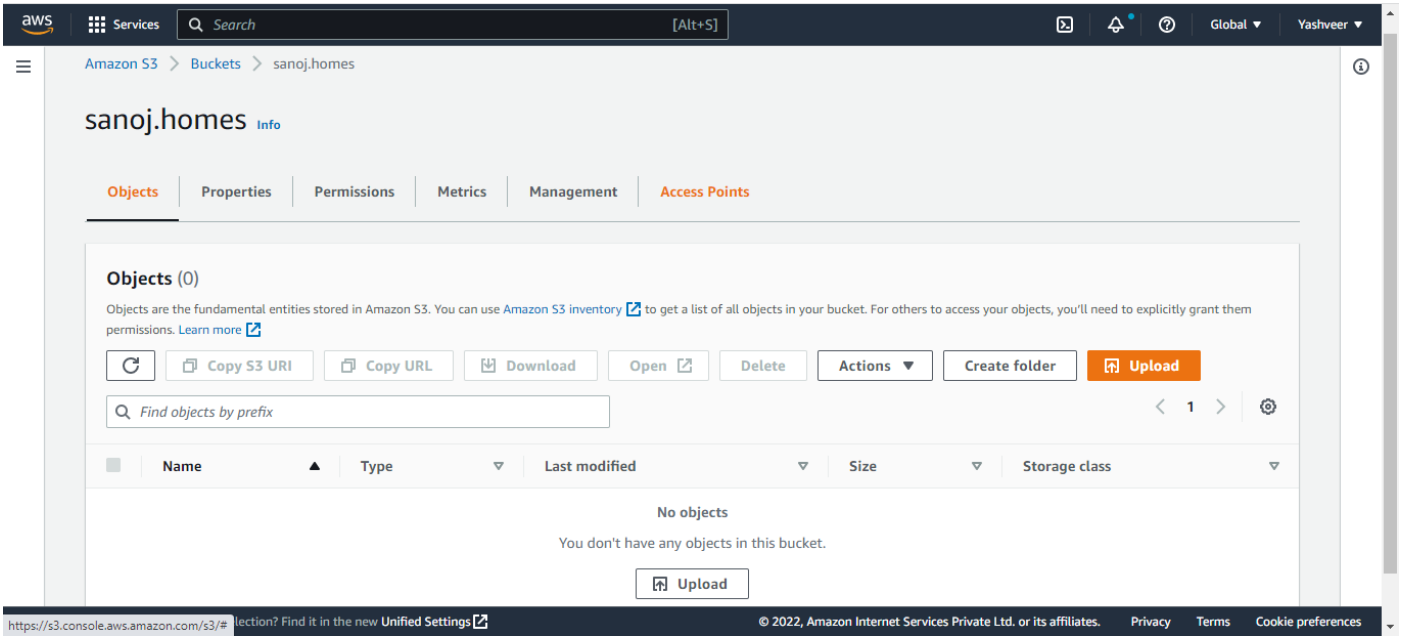
Step3: Now click on the create bucket after creating bucket you will see below interface.



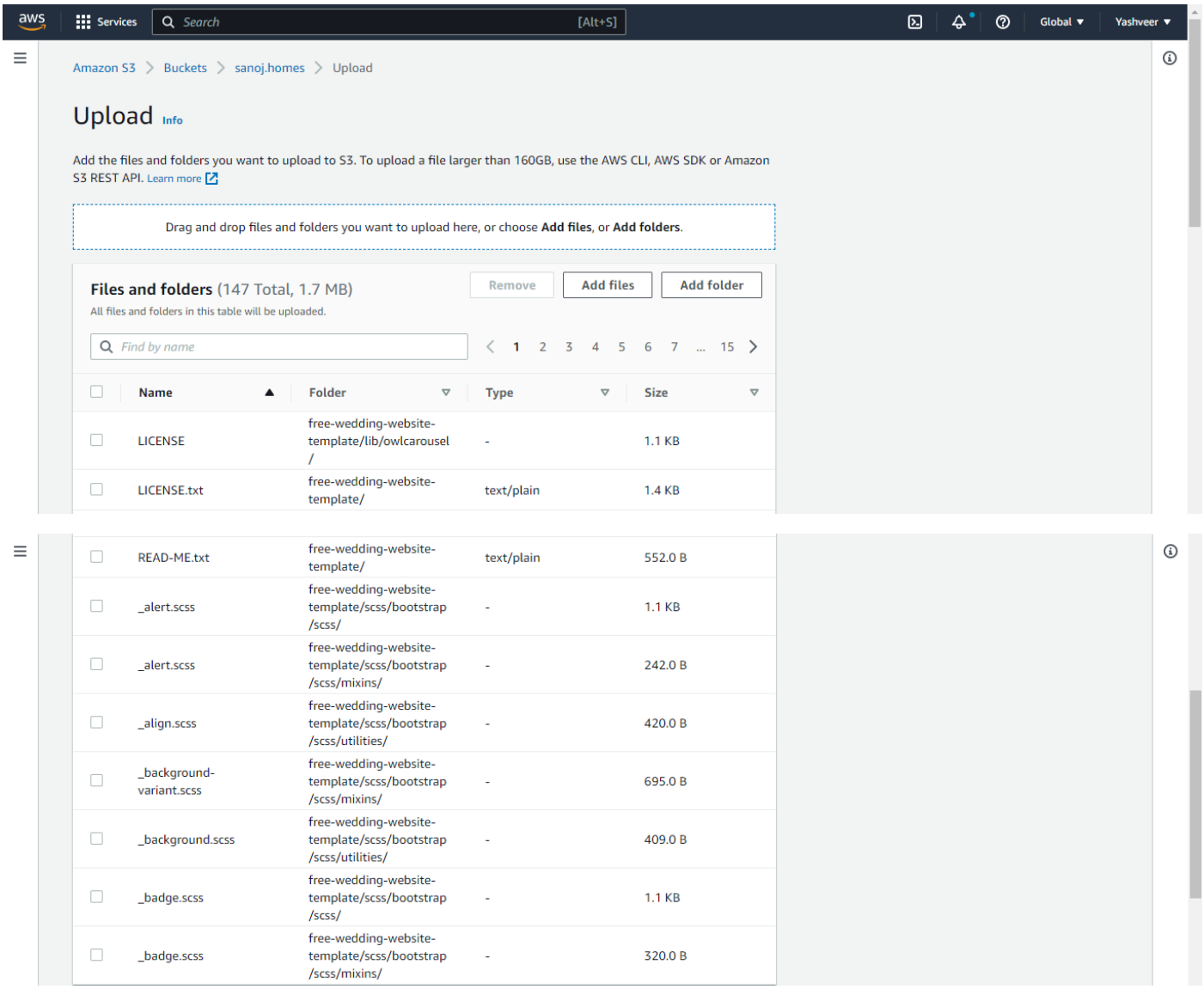
Step 4: Now click on your bucket name

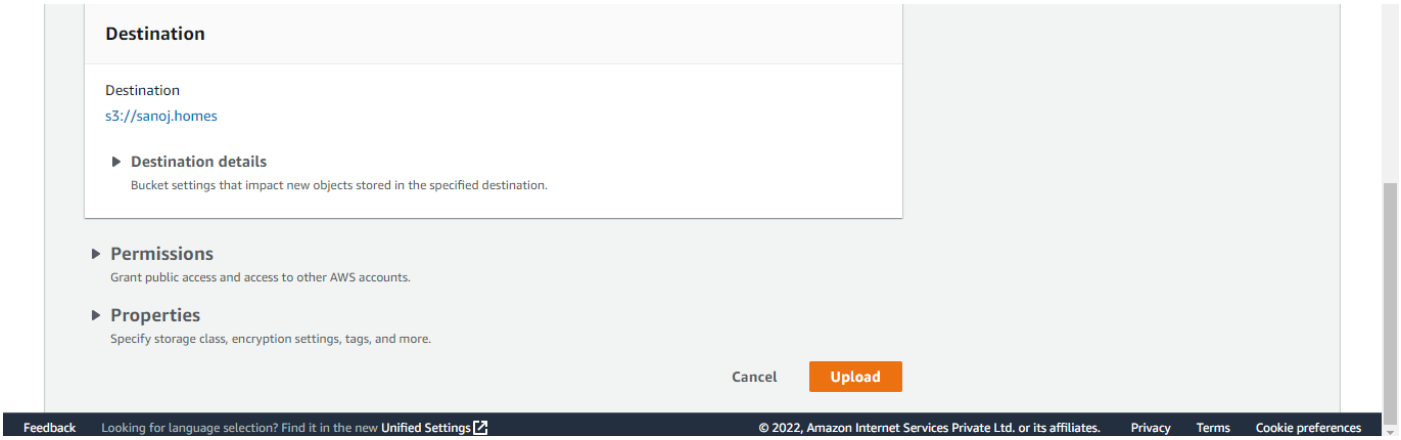


After clicking on the bucket name, you will see below screen.

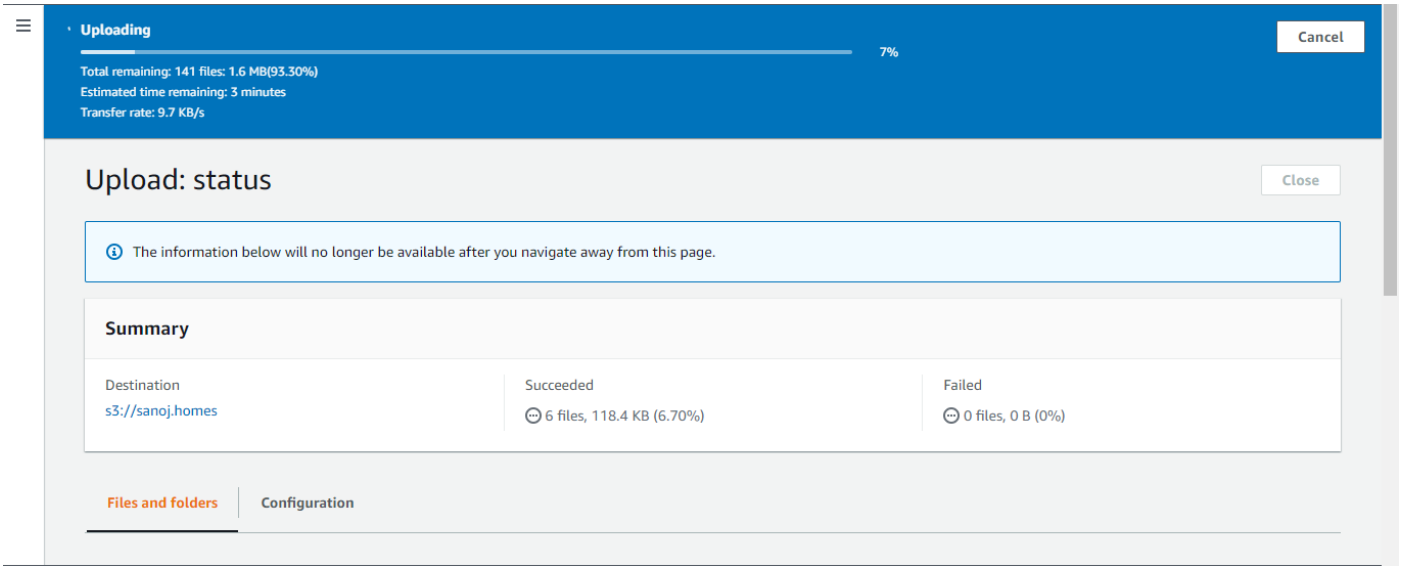


Step 5: Now you have to upload your website on above screen, or you can also drag the file from file manager(File explorer) or drop on the S3 Upload screen after uploading the website content you can see below screen.

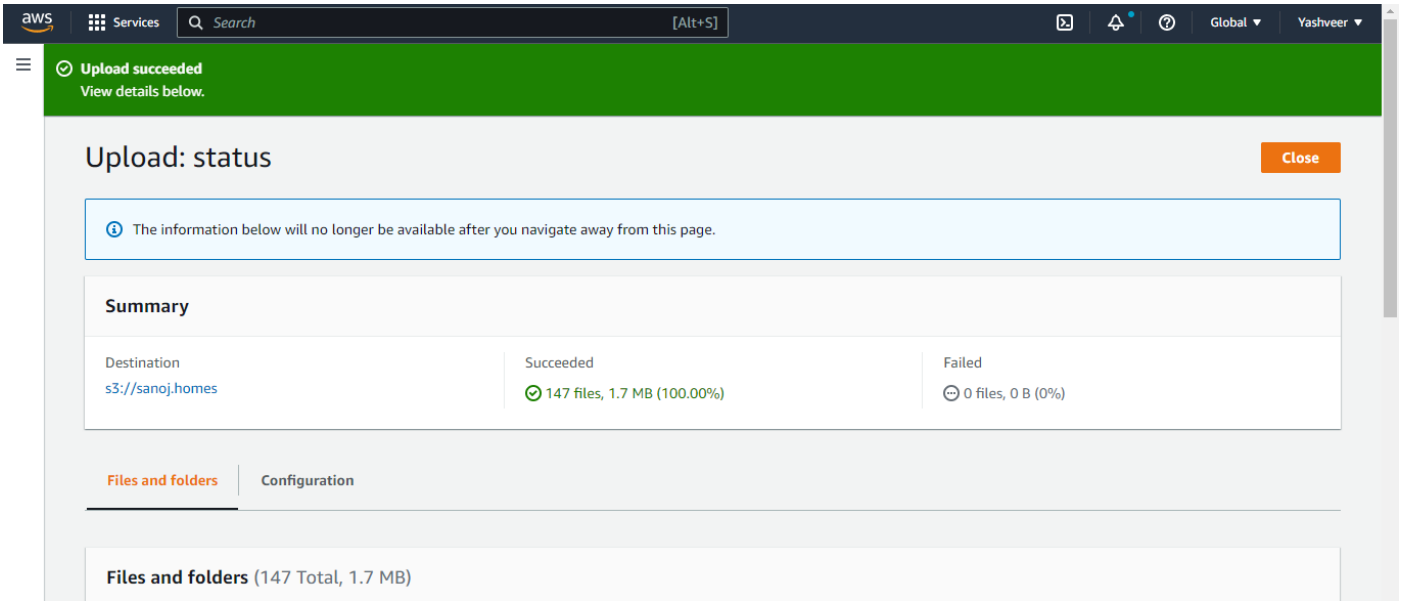




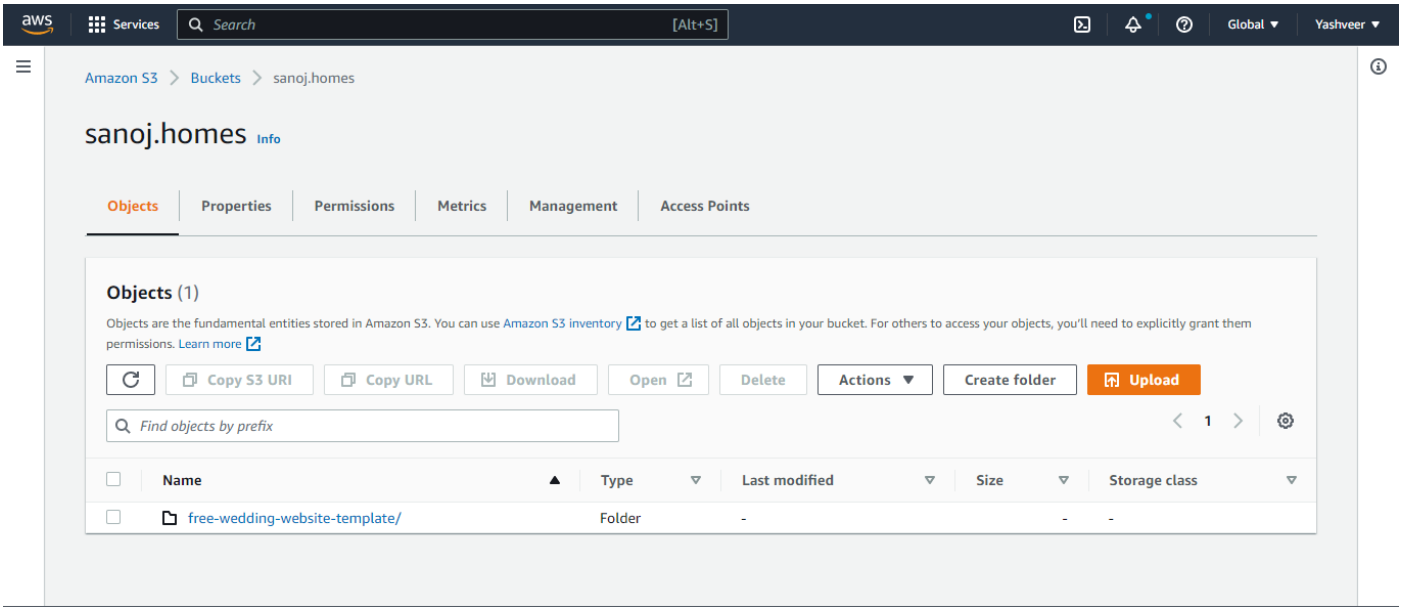
Step 6: Now you have to click on **Upload** after uploading you will see below screen wait till all files upload on S3 bucket.



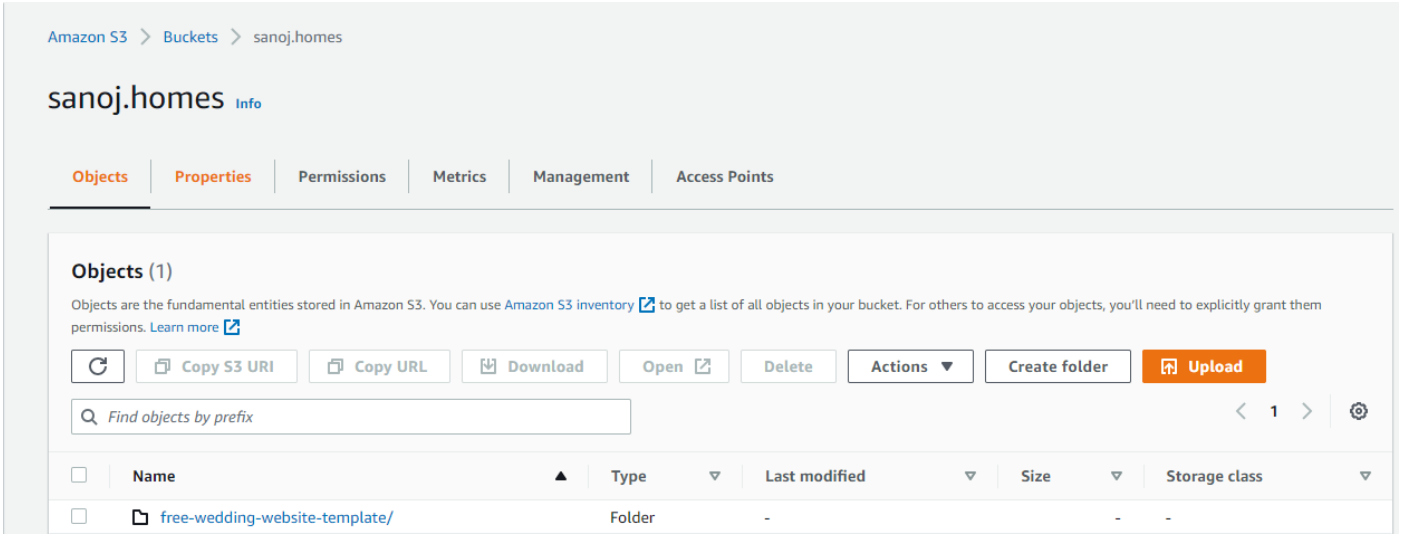
After uploading it will show below kind of interface.



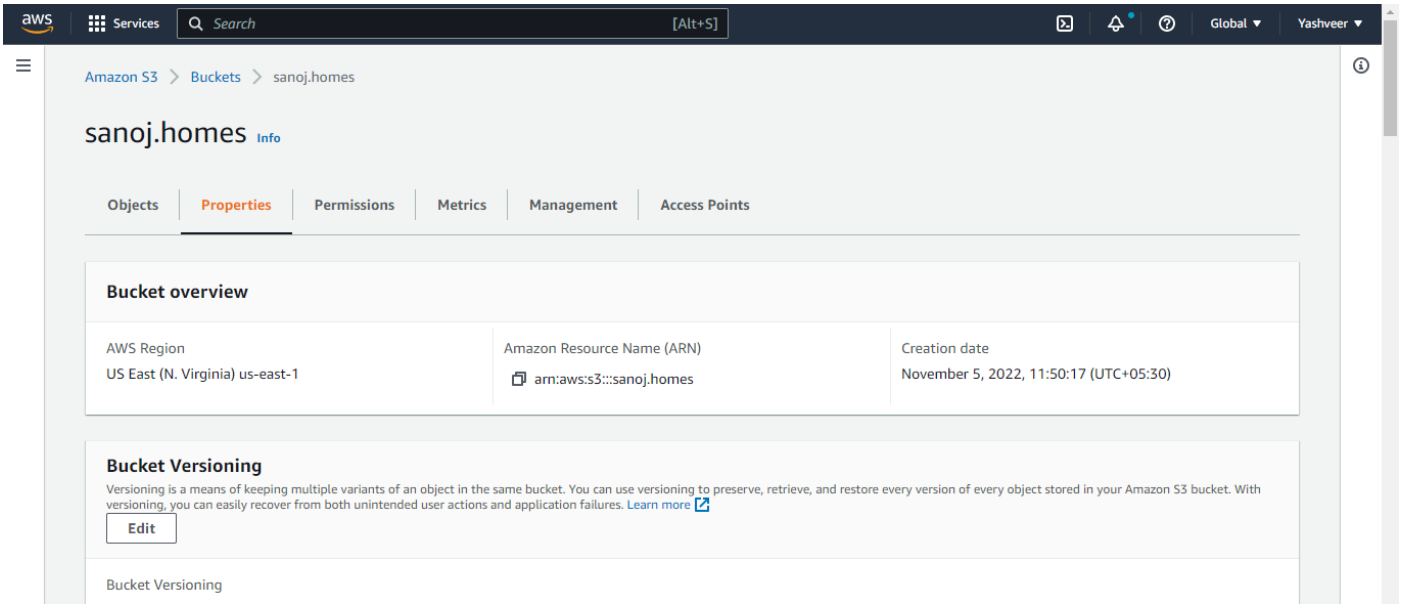
Now you close the window by clicking on **close** button, after that you will see below screen



Step 7: Now click on the **Properties** tab and scroll down till last page of the screen, for reference you can see below screen.



After clicking the Properties tab you will se below interface now scroll down this page till last.



Disabled

Multi-factor authentication (MFA) delete

An additional layer of security that requires multi-factor authentication for changing Bucket Versioning settings and permanently deleting object versions. To modify MFA delete settings, use the AWS CLI, AWS SDK, or the Amazon S3 REST API. Learn more

Disabled

Tags (0)

Edit

Track storage cost or other criteria by tagging your bucket. Learn more

Key	Value

No tags associated with this resource.

Default encryption

Edit

Automatically encrypt new objects stored in this bucket. Learn more

Default encryption

Disabled

Intelligent-Tiering Archive configurations (0)

Enable objects stored in the Intelligent-Tiering storage class to tier-down to the Archive Access tier or the Deep Archive Access tier which are optimized for objects that will be rarely accessed for long periods of time. Learn more

View details | Edit | Delete | Create configuration

Find Intelligent-Tiering Archive configurations

Name	Status	Scope	Days until transition to Archive Access tier	Days until transition to Deep Archive Access tier

No archive configurations

No configurations to display.

Create configuration

Server access logging

Edit

Log requests for access to your bucket. Learn more

Server access logging

Disabled

AWS CloudTrail data events

Configure in CloudTrail

Configure CloudTrail data events to log Amazon S3 object-level API operations in the CloudTrail console. Learn more

Name	Access

No data events

No data events to display.

Configure in CloudTrail

Event notifications (0)

Edit | Delete | Create event notification

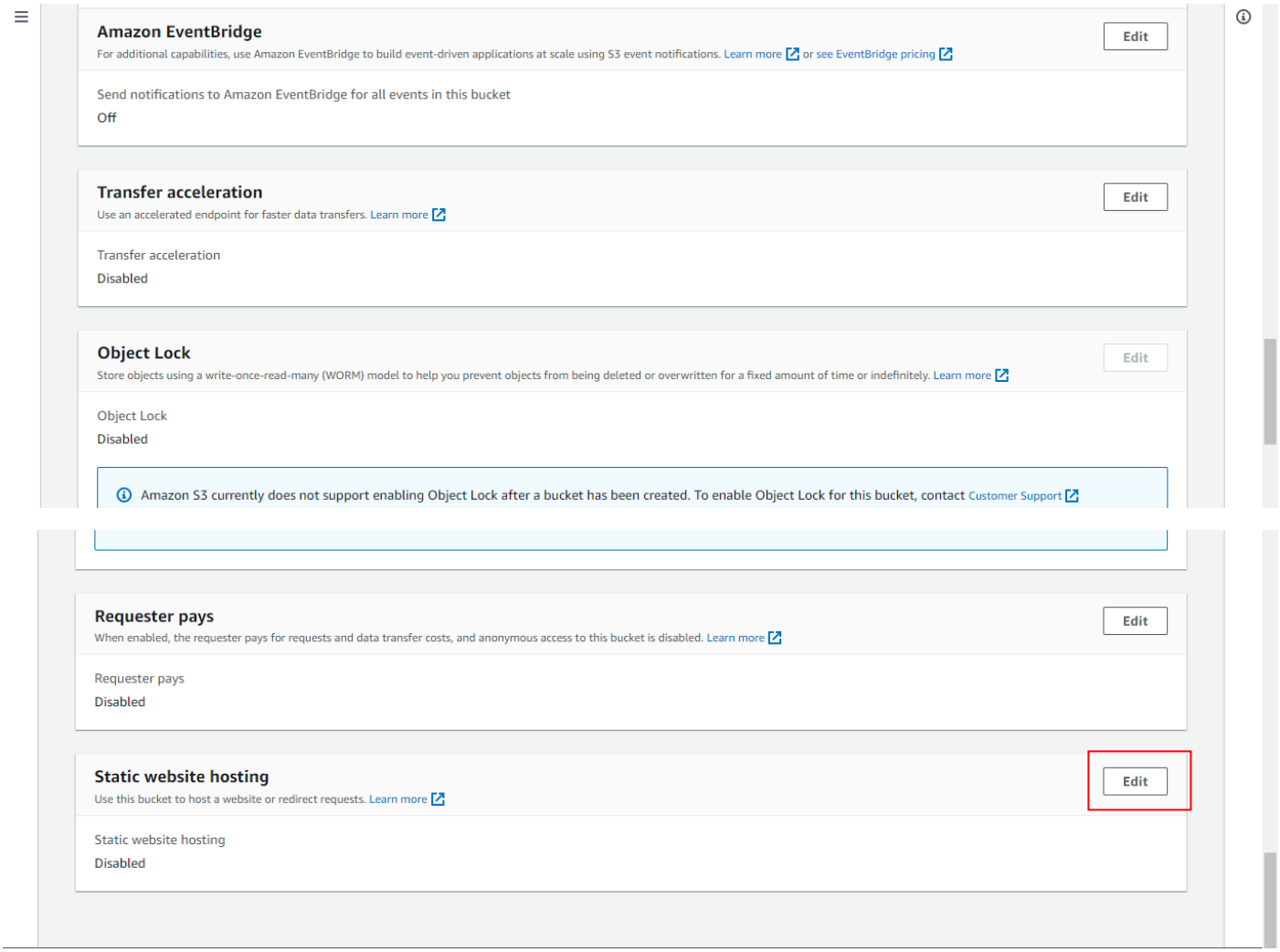
Send a notification when specific events occur in your bucket. Learn more

Name	Event types	Filters	Destination type	Destination

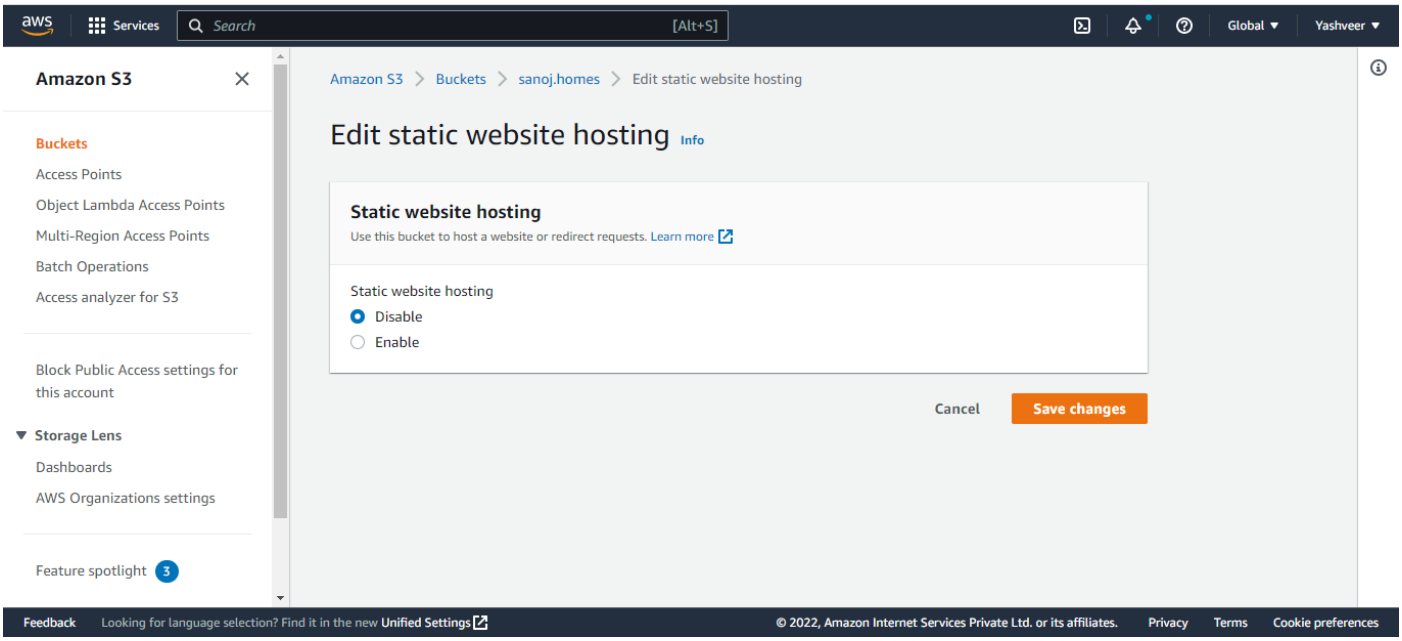
No event notifications

Choose **Create event notification** to be notified when a specific event occurs.

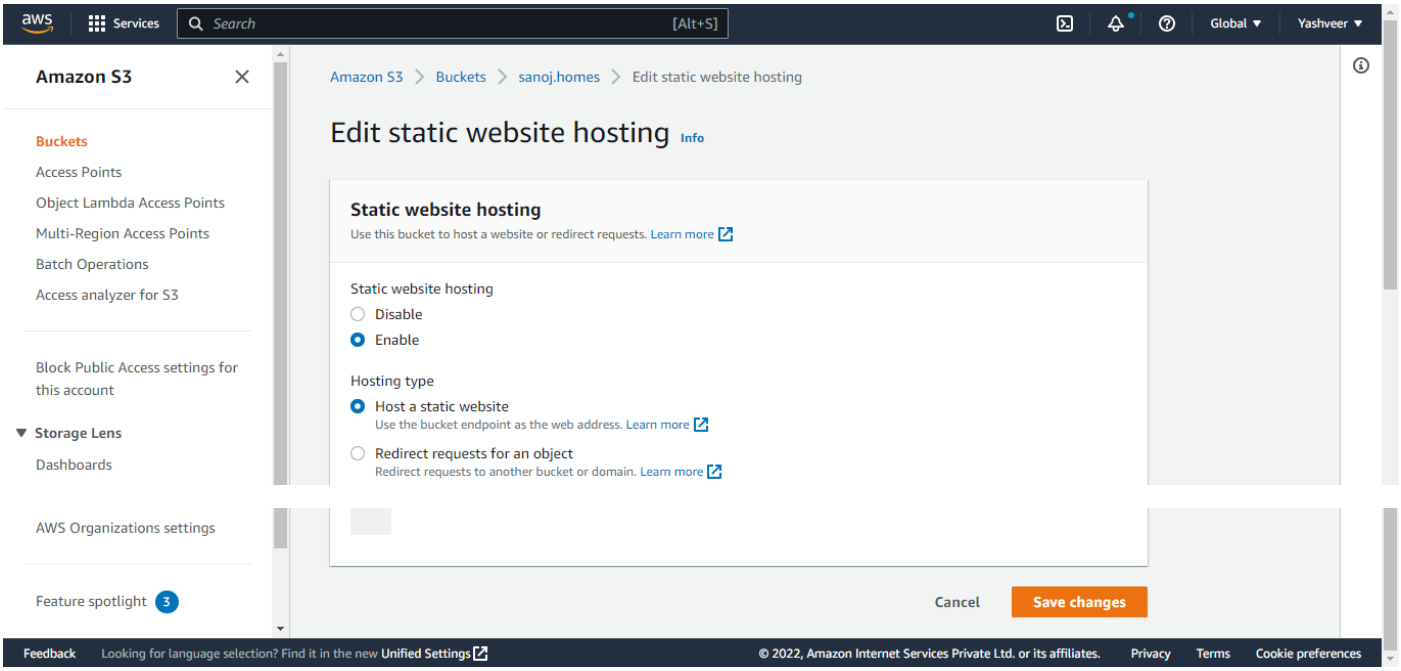
Create event notification



Step 8: Now as you can see in the above image **"Static website hosting"** click on the **"Edit"** after clicking on the edit you will see below screen.

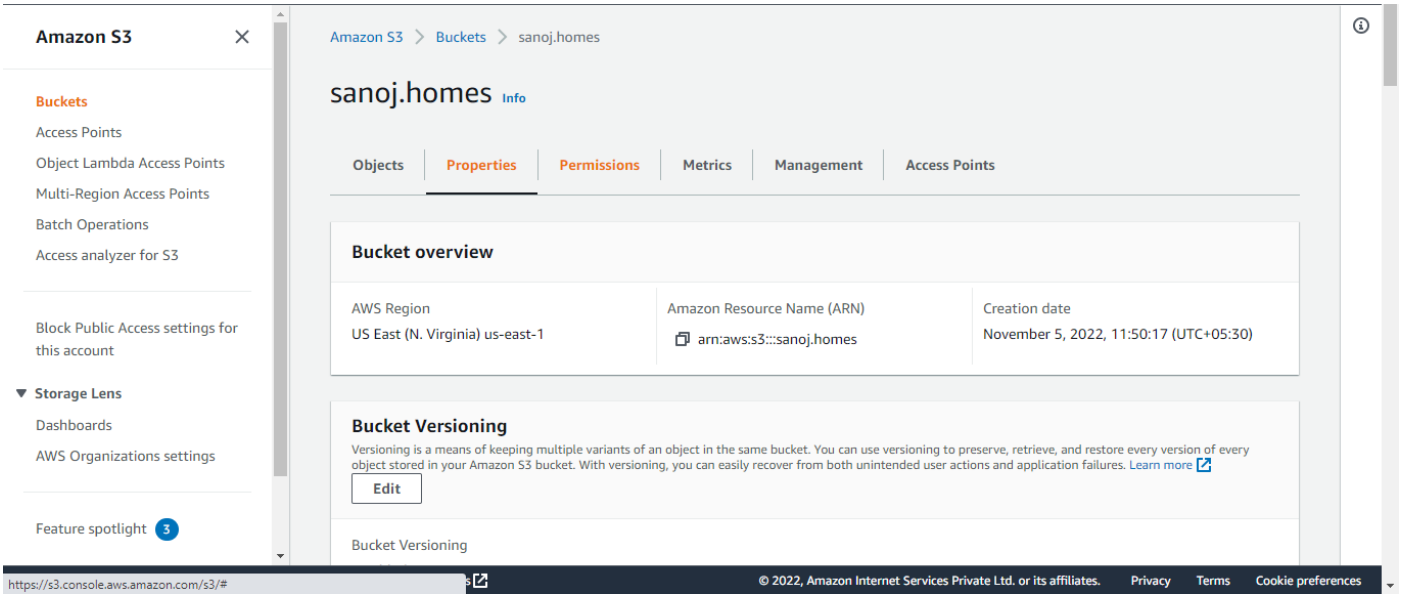


Step 9: Now click on the **"Enable"**.

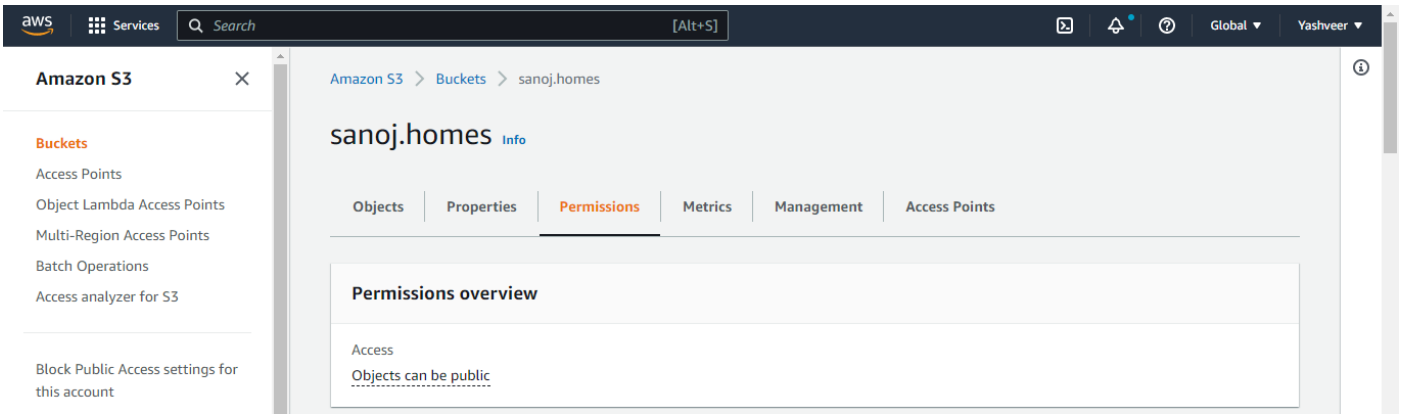


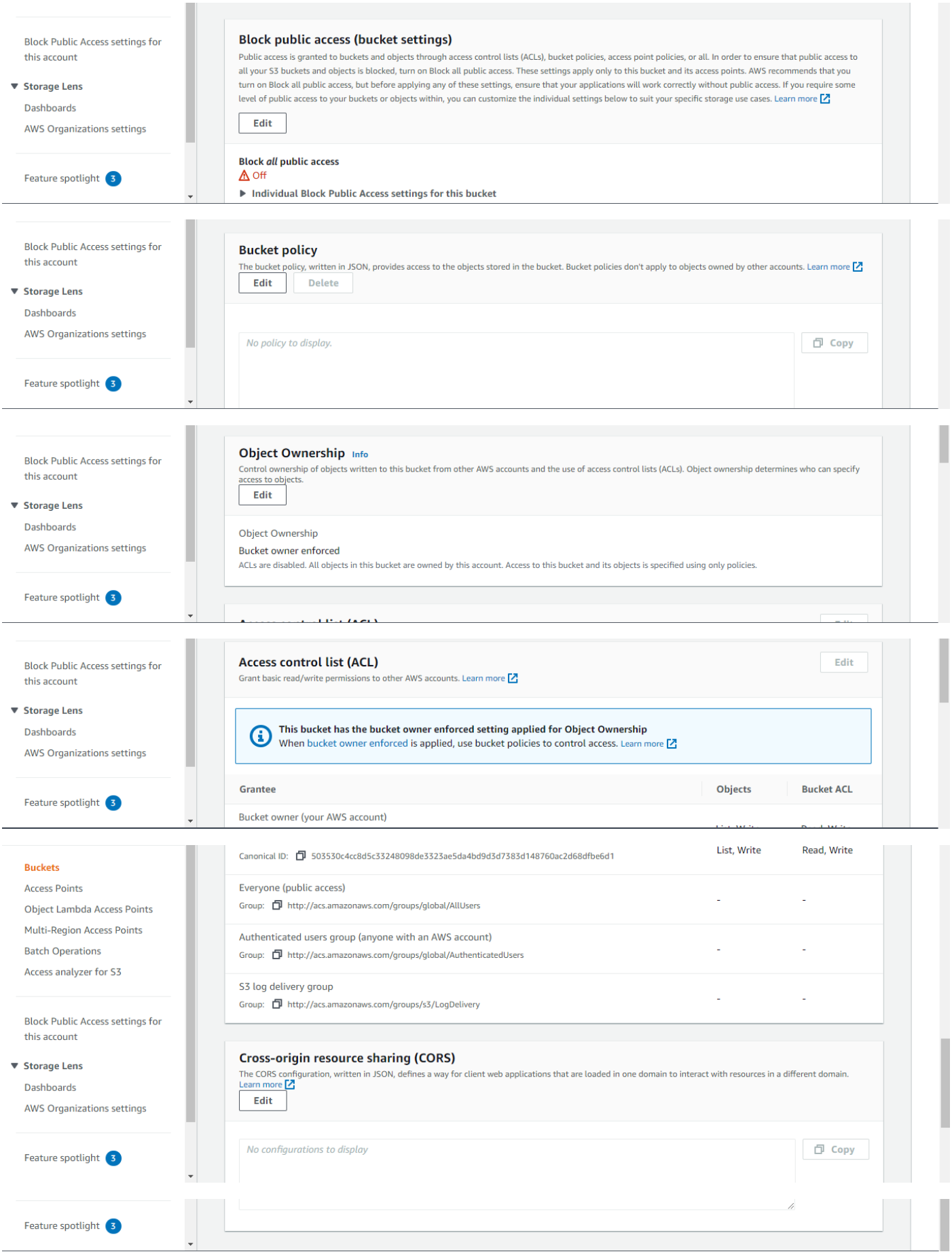
Now click on **"Save changes"**

Step 10: Click on **"Permission"** and applied bucket policy for reference you can see below image.



After Click on the "Properties" tab you will see below kind of interface.

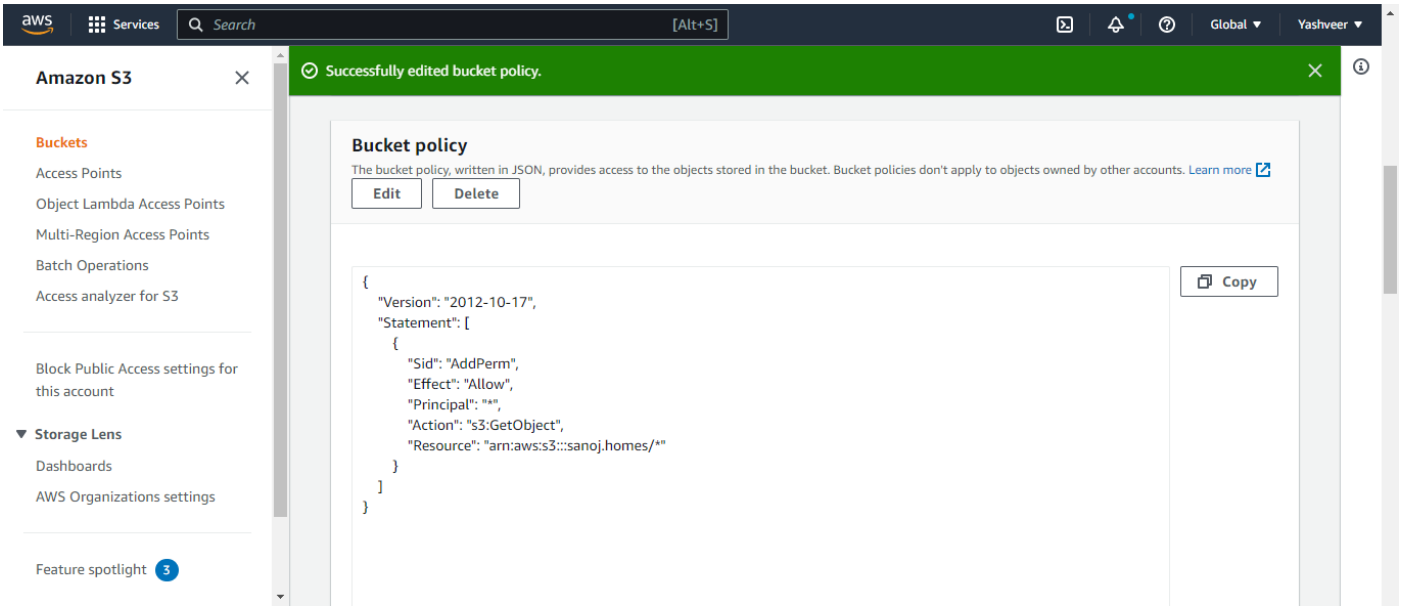




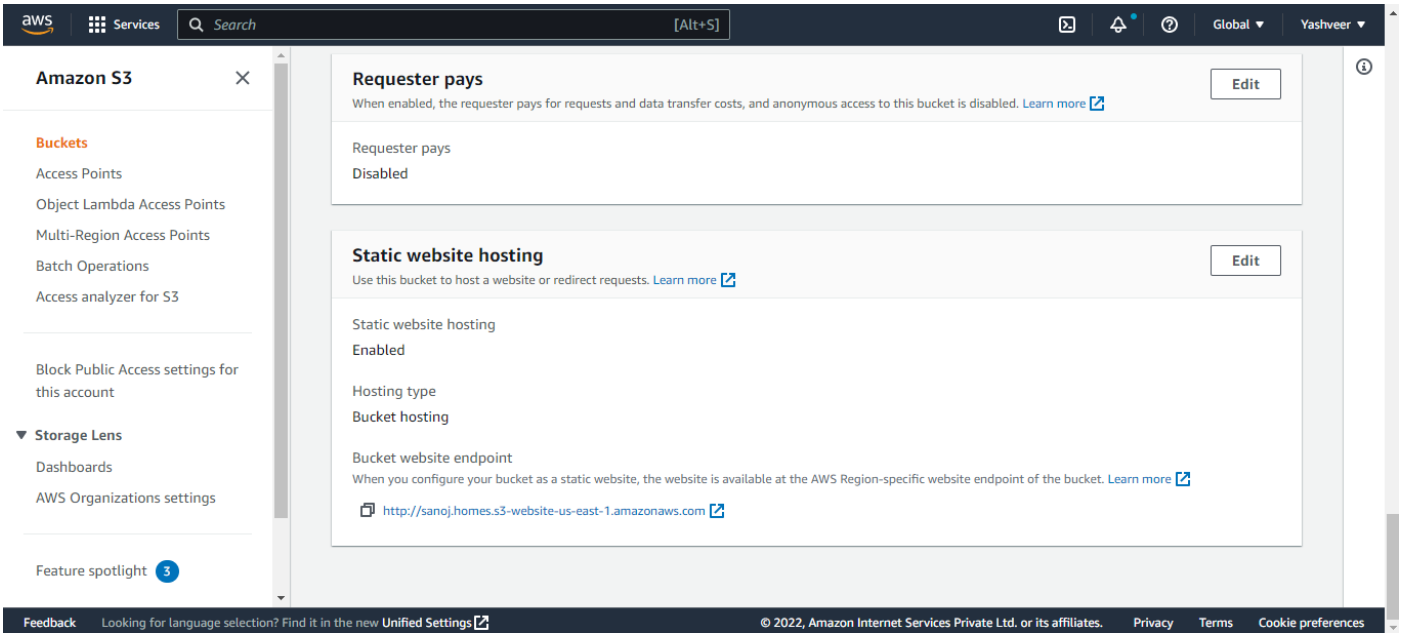
Step 11: Now we have to apply the below mentioned policy on the bucket so it can be accessible publicly.
By clicking on the **"Edit"** to edit Bucket policy

Policy:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AddPerm",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::sanoj.home/*"
    }
  ]
}
```



Step 12: Now click on the “properties” tab and scroll down the page till last and copy the URL. for refrence you can see below image.



Now you have completed the S3 part, you can try to hit S3 URL like below:

http://sanoj.home.s3-website.ap-south-1.amazonaws.com/free-wedding-website-template/index.html

Syntax of Link: <s3 bucket URL> <inside bucket folder Name> <website page name>

Here

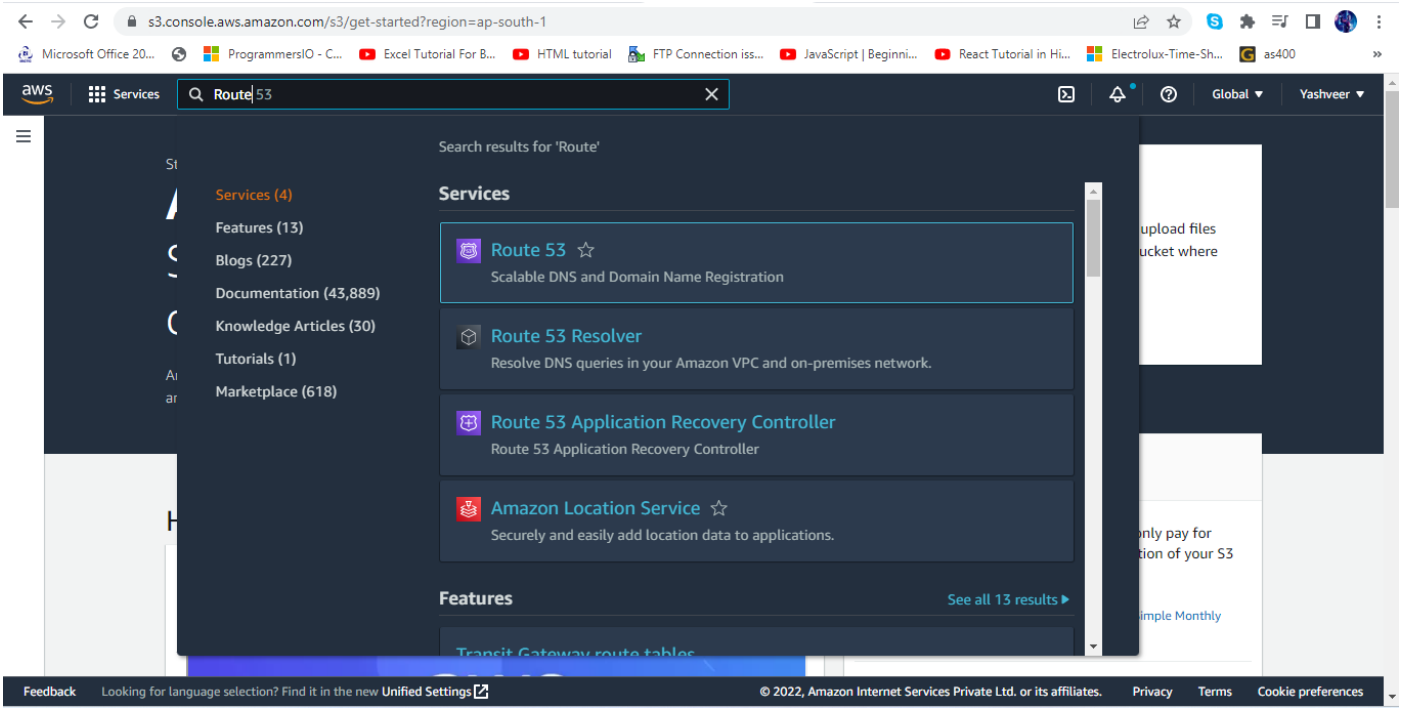
This is **S3 bucket URL:** http://sanoj.home.s3-website.ap-south-1.amazonaws.com

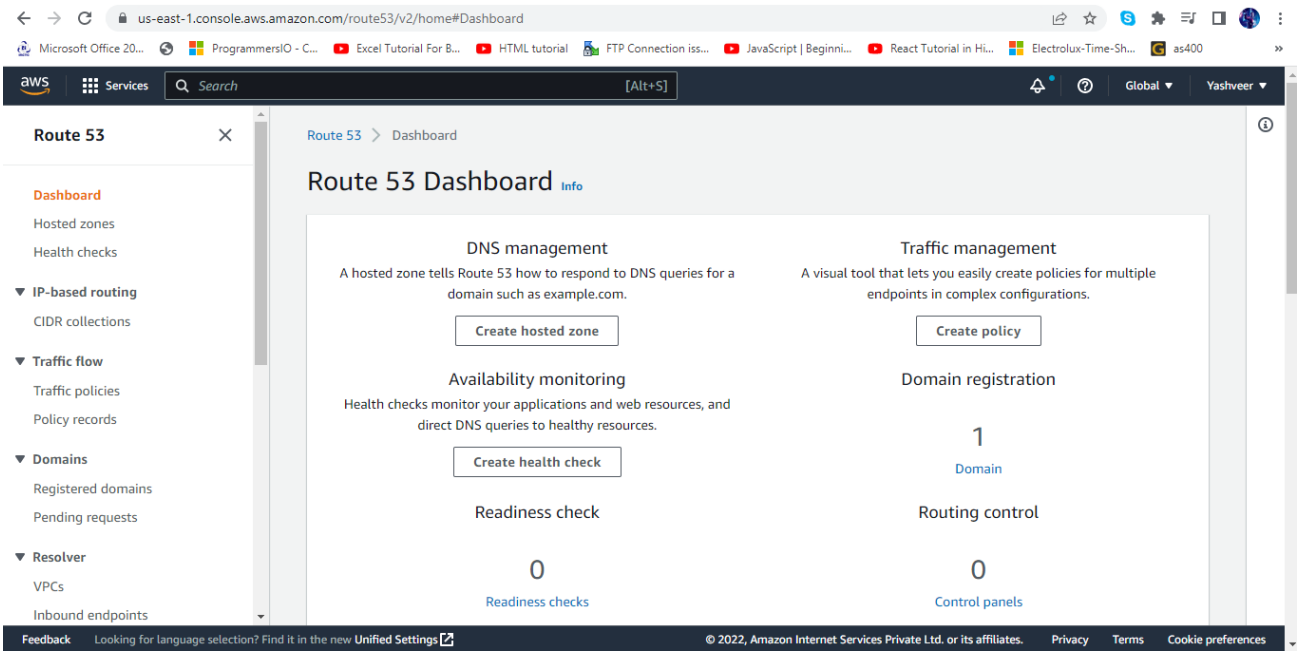
This is **Inside bucket Folder Name**: free-wedding-website-template

This is **website main page name**: index.html

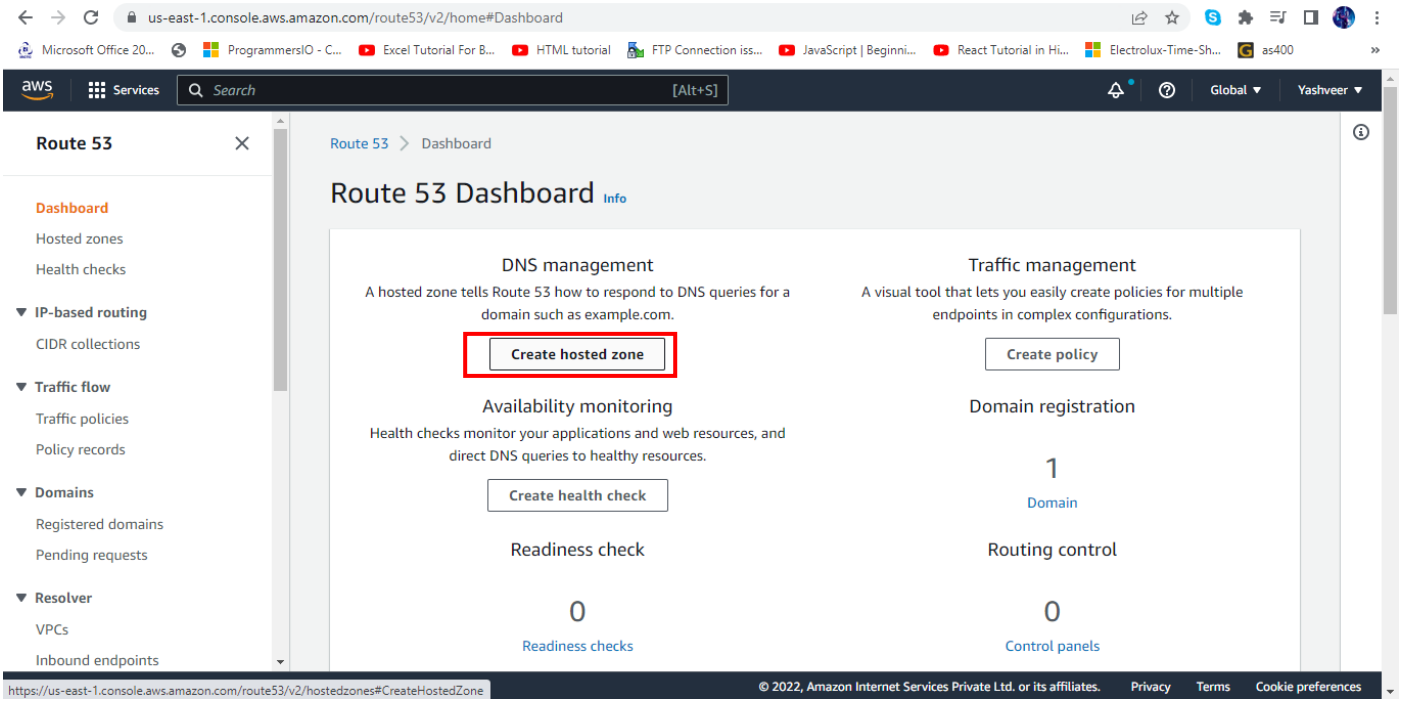
Setting Up Route 53:

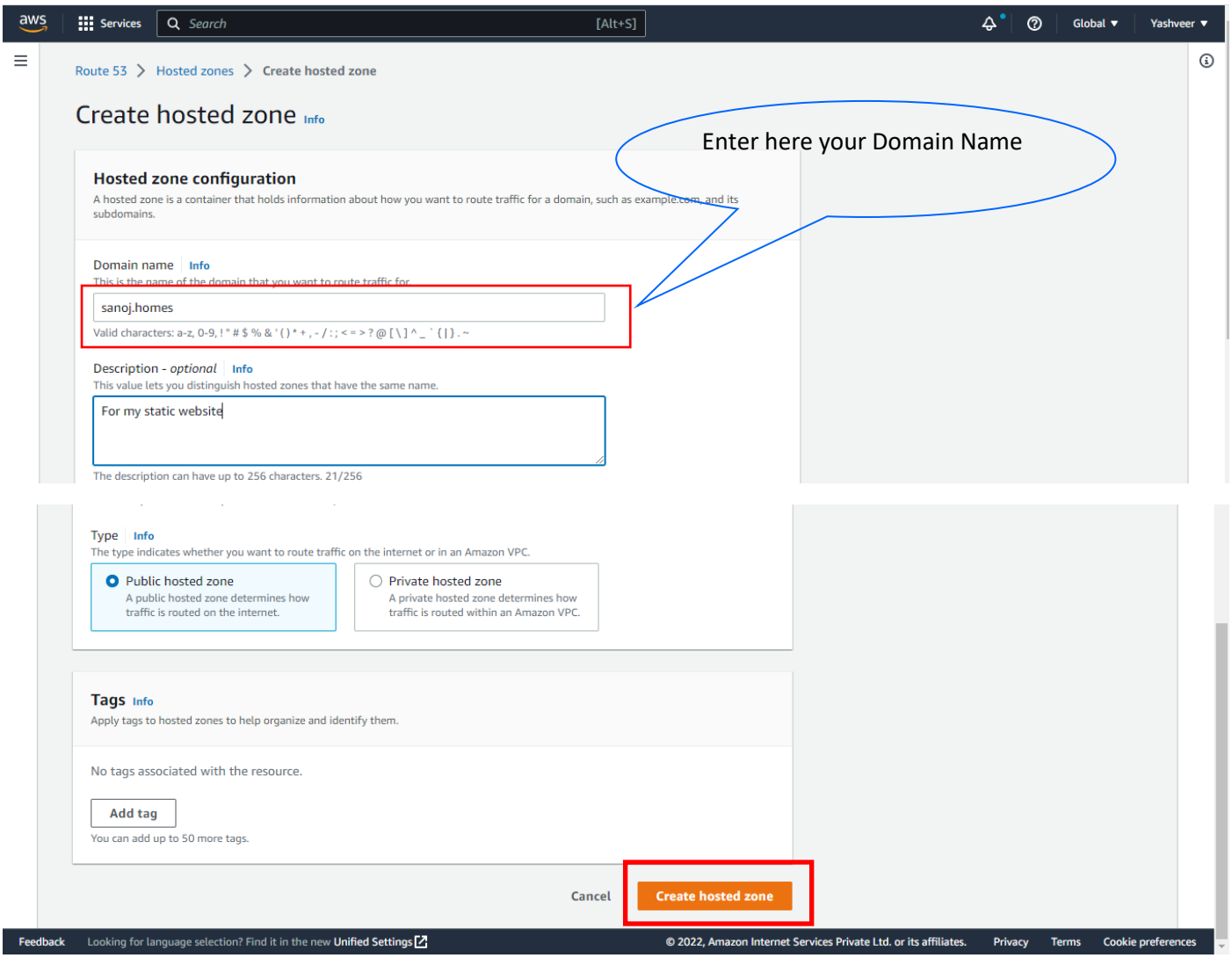
Step1: In the AWS console just search for Route 53 and hit enter you will see below type of interface , after search hit enter.



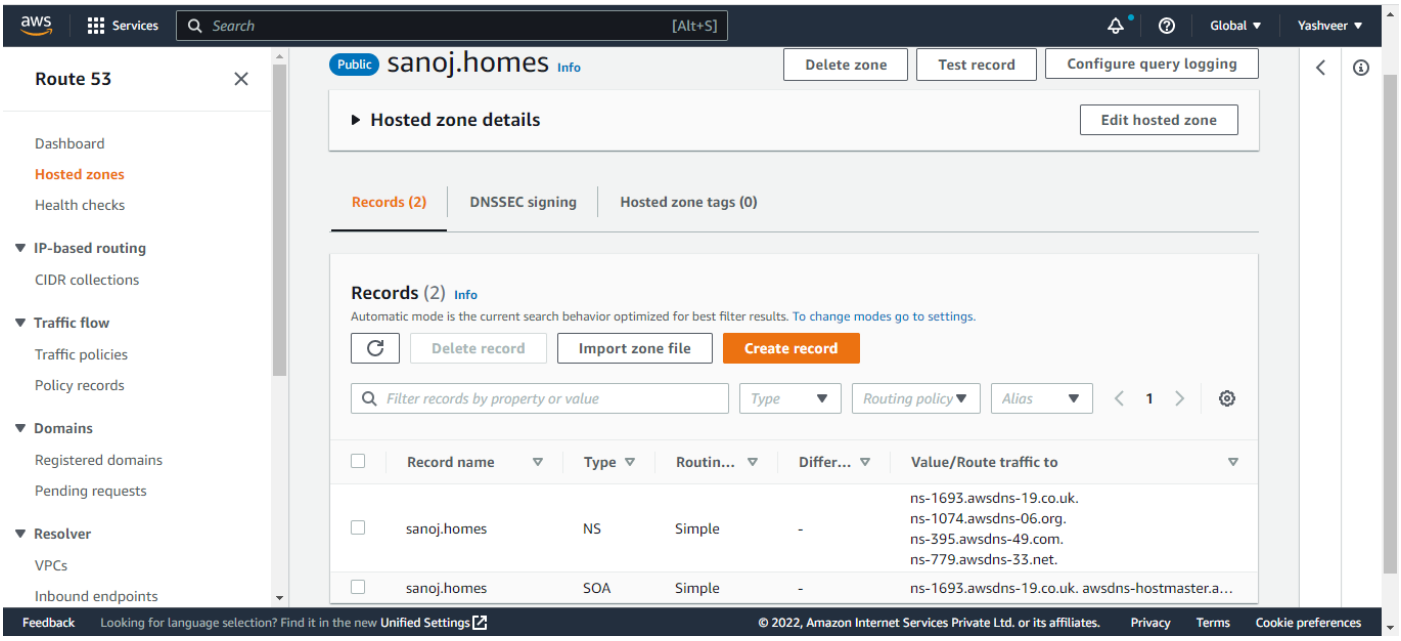


Step 2: Now you have to create a hosted zone, so just click on the **"create hosted zone"** it will redirect to below kind of interface.

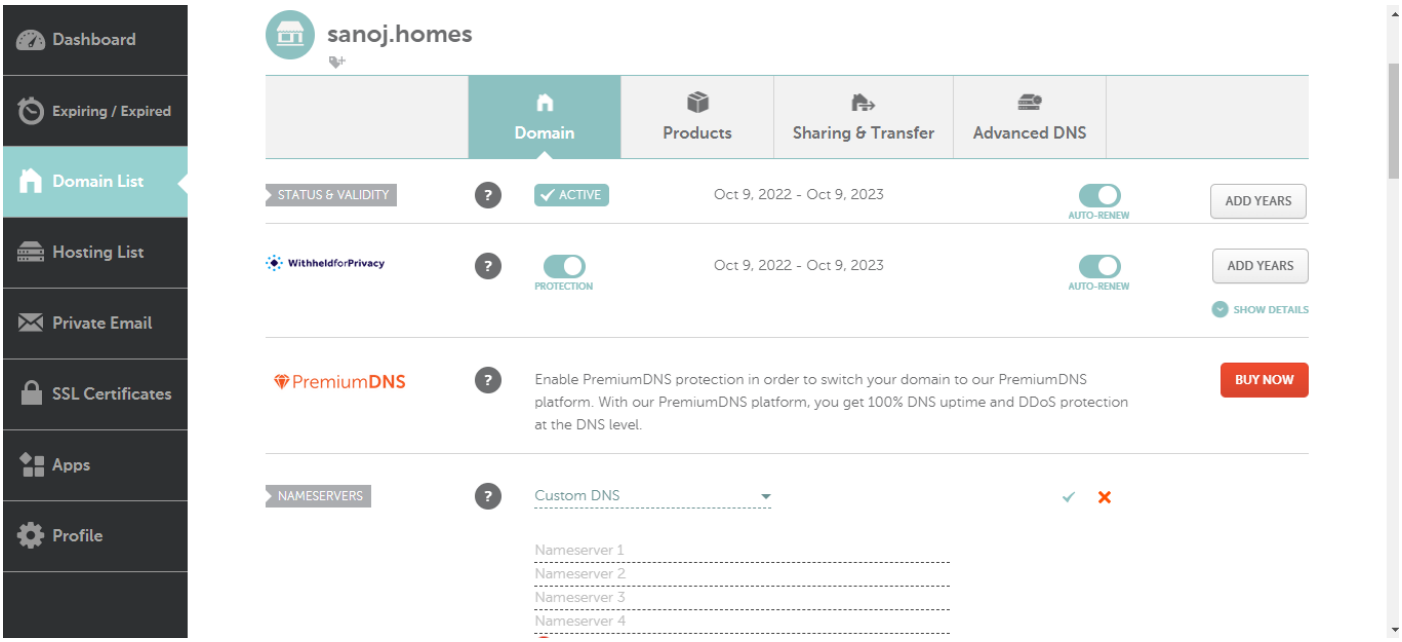




Now you have to just click on the **"Create hosted zone"** button. after click on the Create hosted zone you will see there is two records is created **Name Server (NS)** and **Source Of Authority(SOA).**

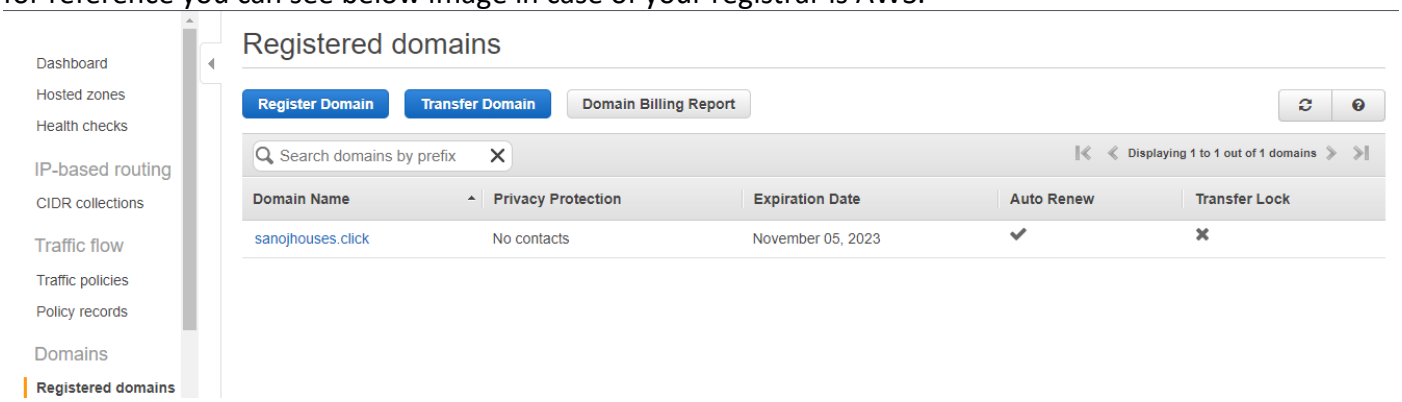


Now you have to update the NS(Name Server) Record on your domain registrar in my case I have **"Namecheap"** in your case might be it is goDaddy, Hostinger, freenom etc.



Step 3: Now go to the Route 53 and Copy the NS(Name Server) Records and paste it here if you purchased your domain from 3rd party , if you purchased domain from AWS you have to update this name server in AWS route 53 **"Registered domains"** section.

for reference you can see below image in case of your registrar is AWS.

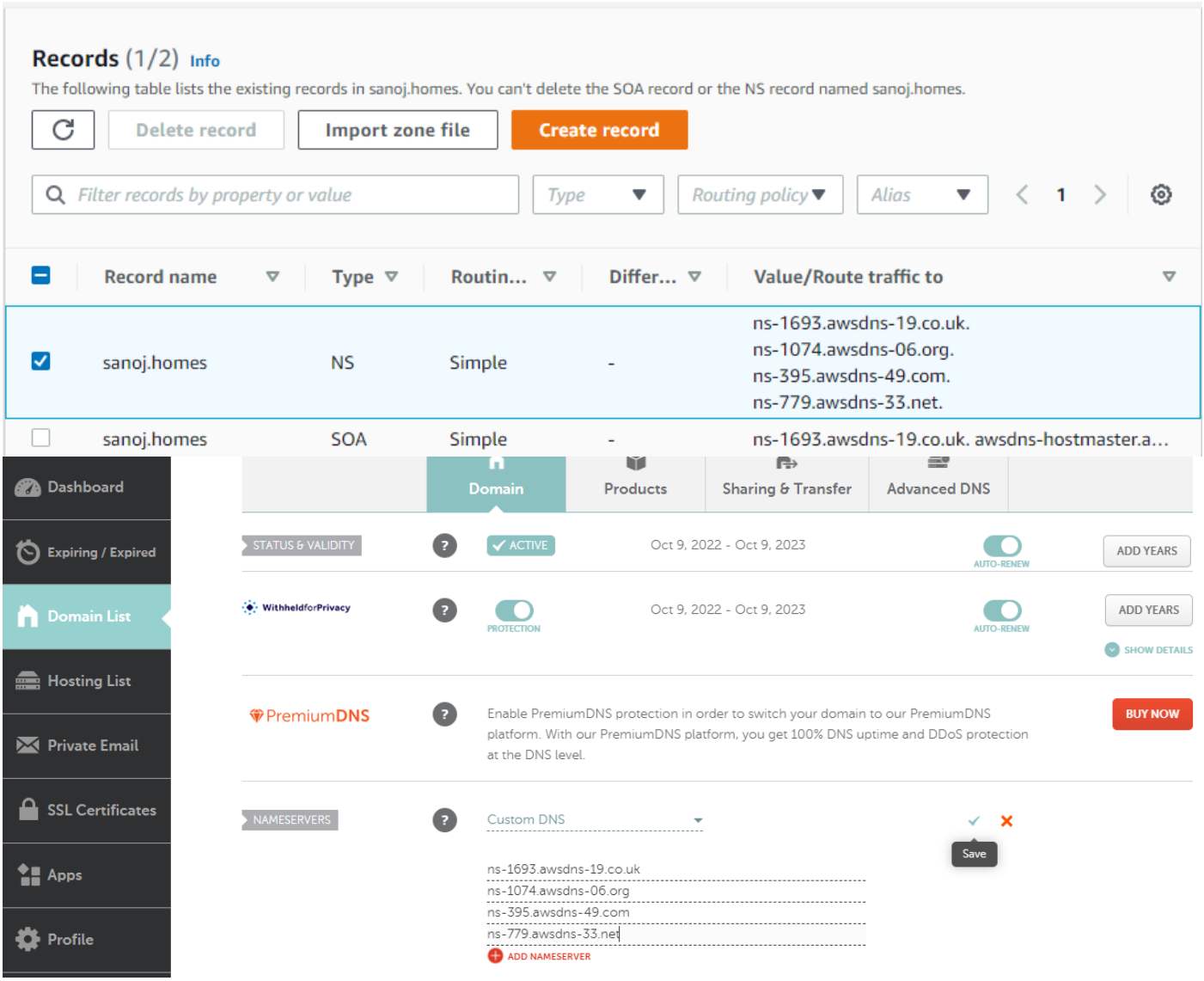


Let’s update the our **“sanoj.homes”** domain records in Namecheap NS records.

Let me tell you behind the scene what happen if we update ns records on Namecheap.

Whenever request is hit for **“sanoj.homes”** it will redirect to “Namecheape” to resolve the DNS name once it will get the NS record then this record is created by route 53 so it is resolved by Route53. after that you will able to see your website where it is hosted.

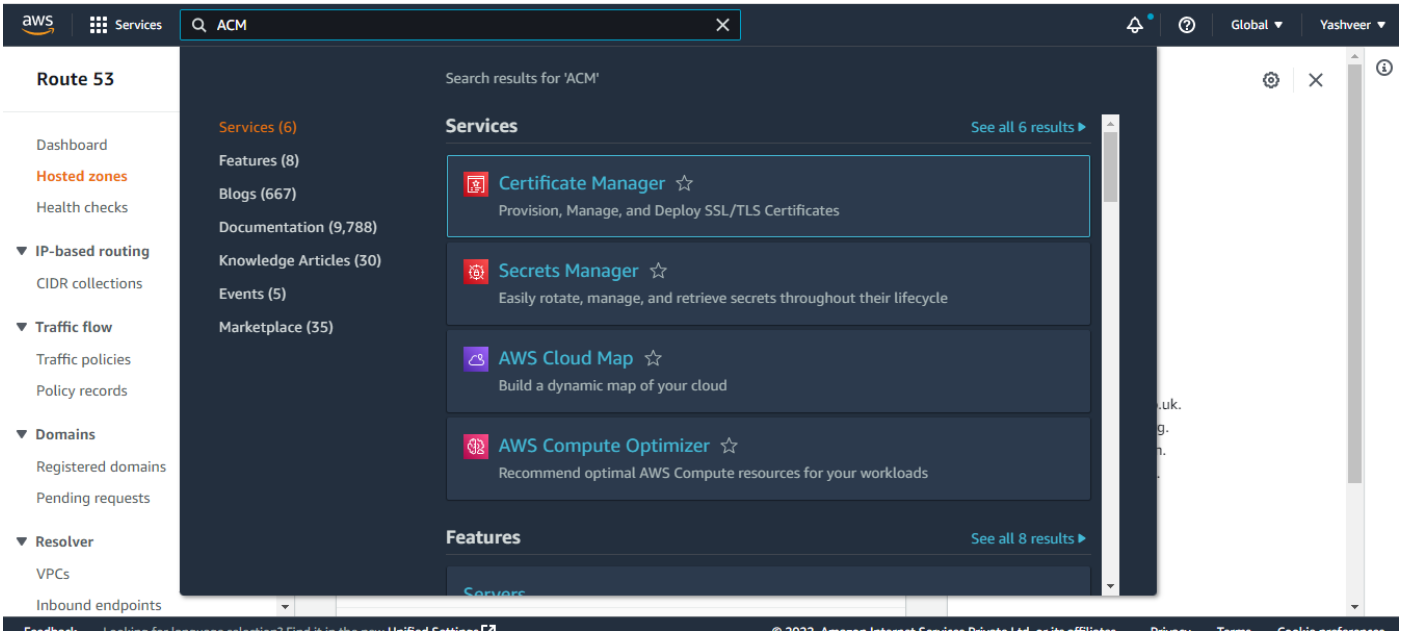
Step 4: Now copying the below NS records from Route 53 and paste it on the Namecheap custom DNS.

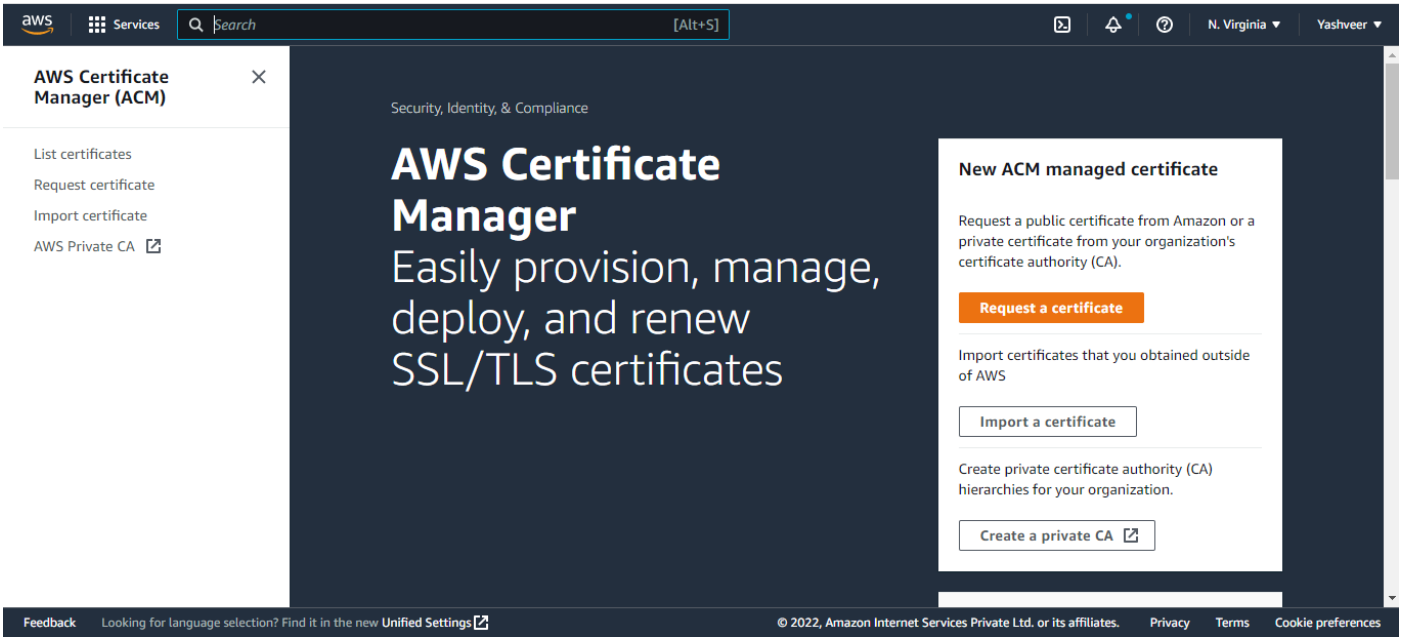


Step 5: After copy and paste just save this NS records on Namecheap.

Setting up Amazon Certificate Manager:

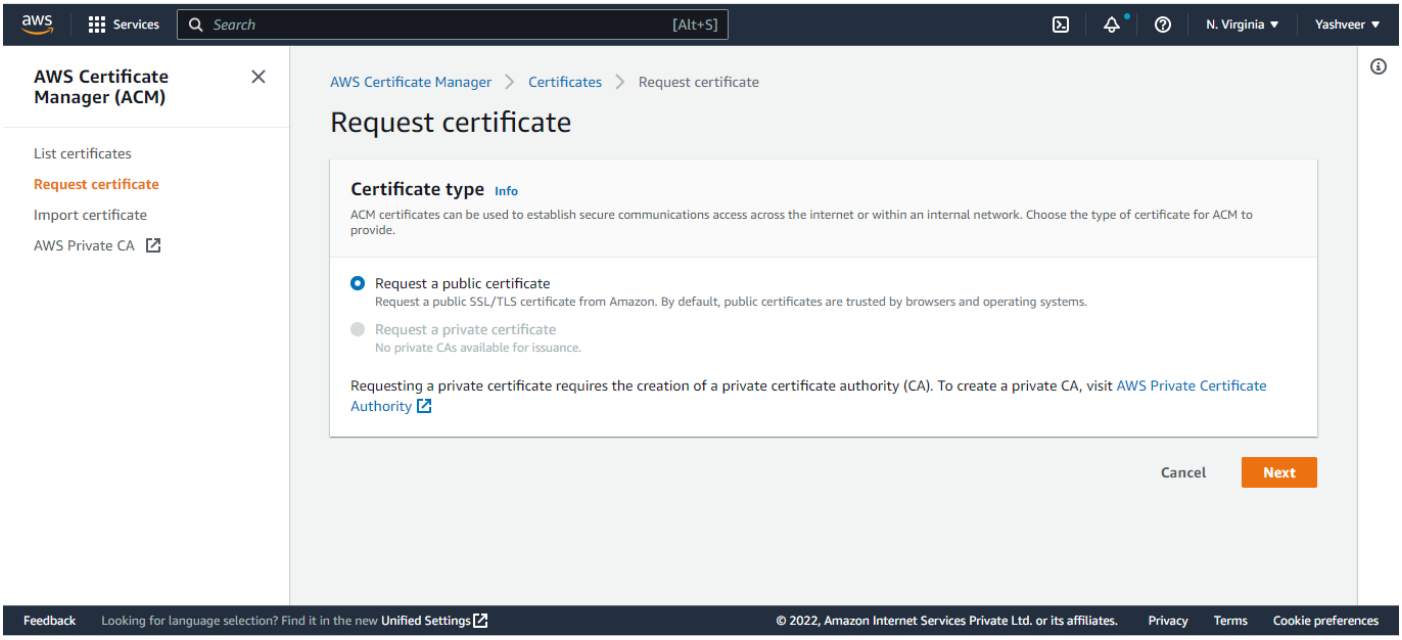
Step1: Search ACM on the Console Search bar and Hit enter you will see below kind of interface. Make sure certificate should be created on the “N.Virginia”. otherwise it will not reflect on your system while you are attaching it into the Cloudfront.



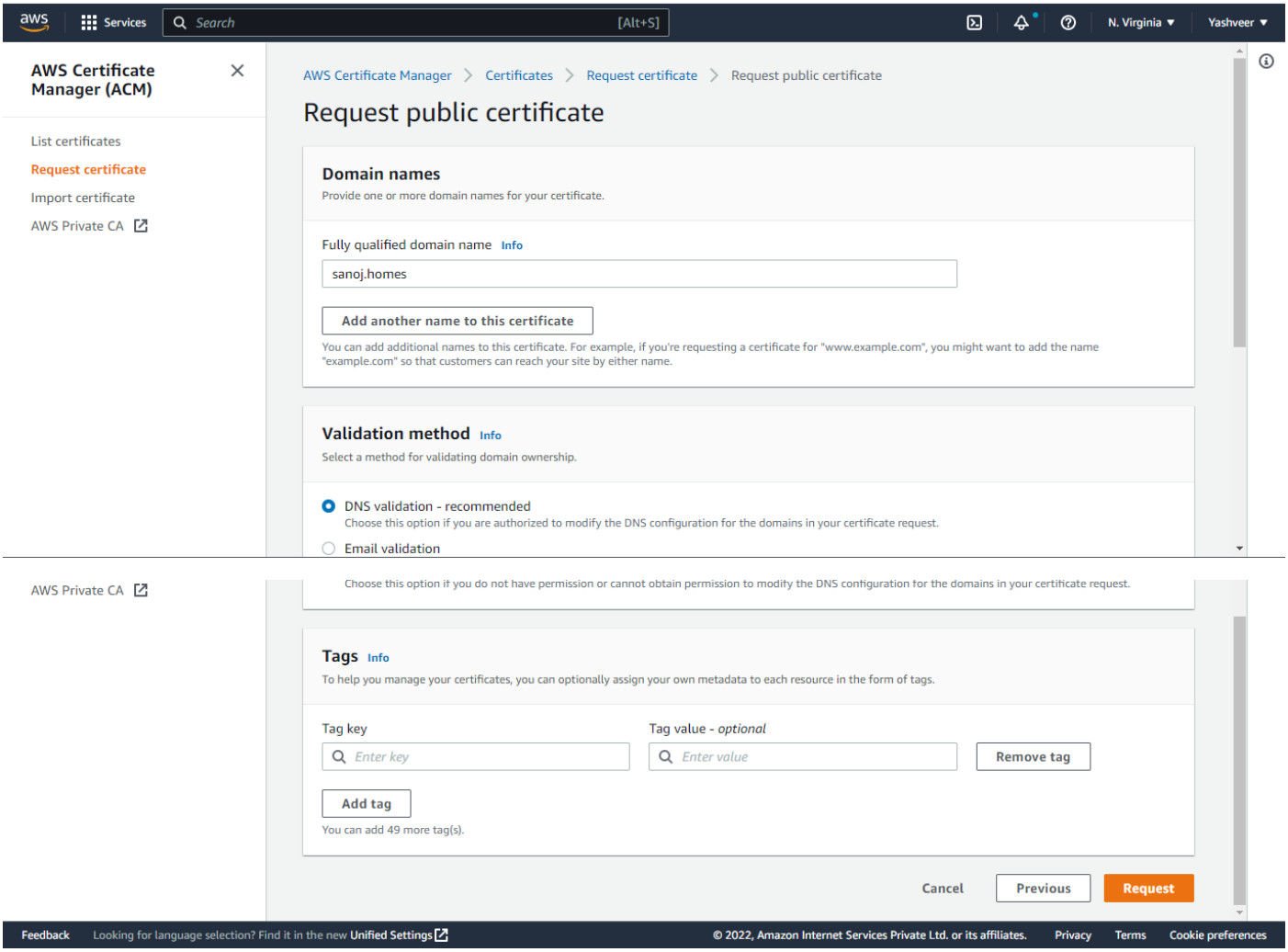


Step 2: Now just click on the **“Request certificate”**

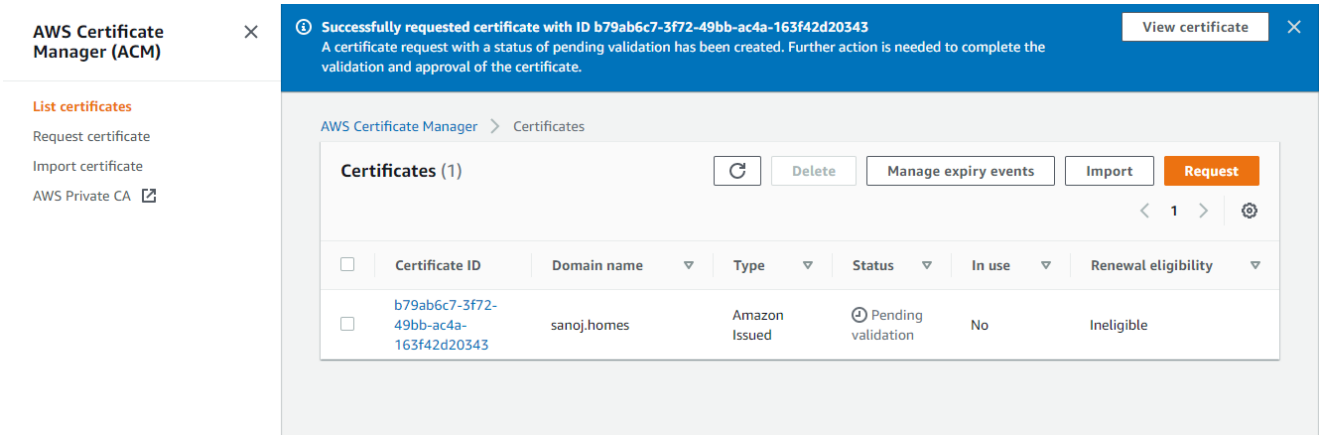
After clicking on the Request Certificate you will see below interface.



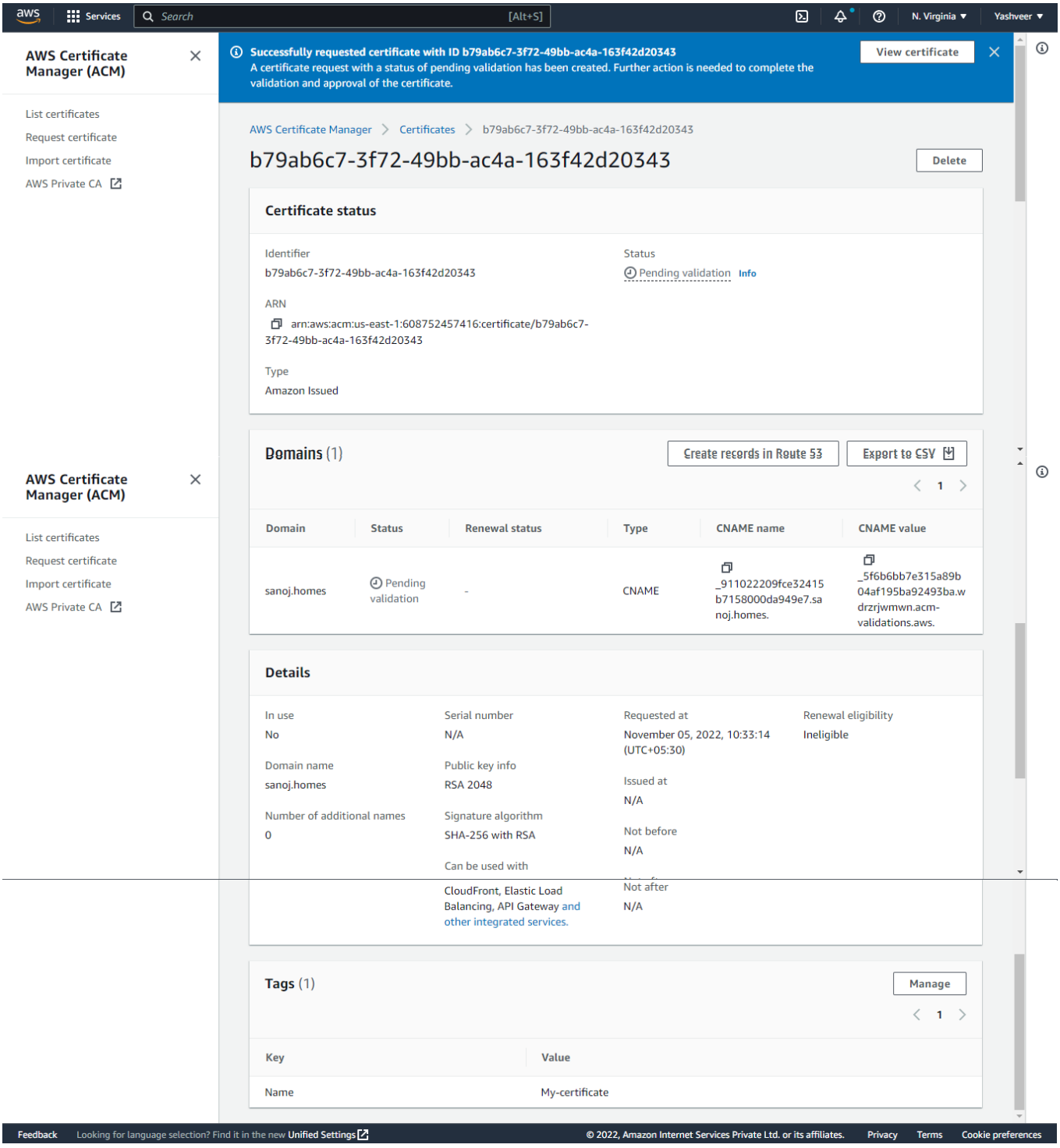
Step 3: Then Click on **Next.**



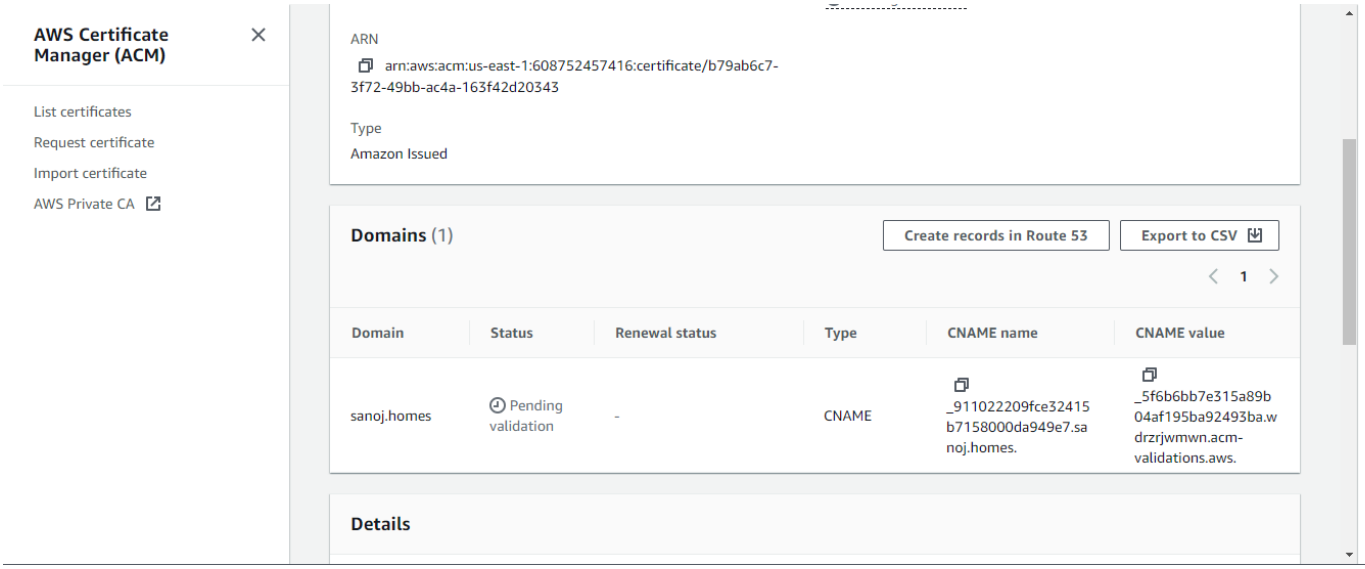
Step 4: Now click on **“Request”.**



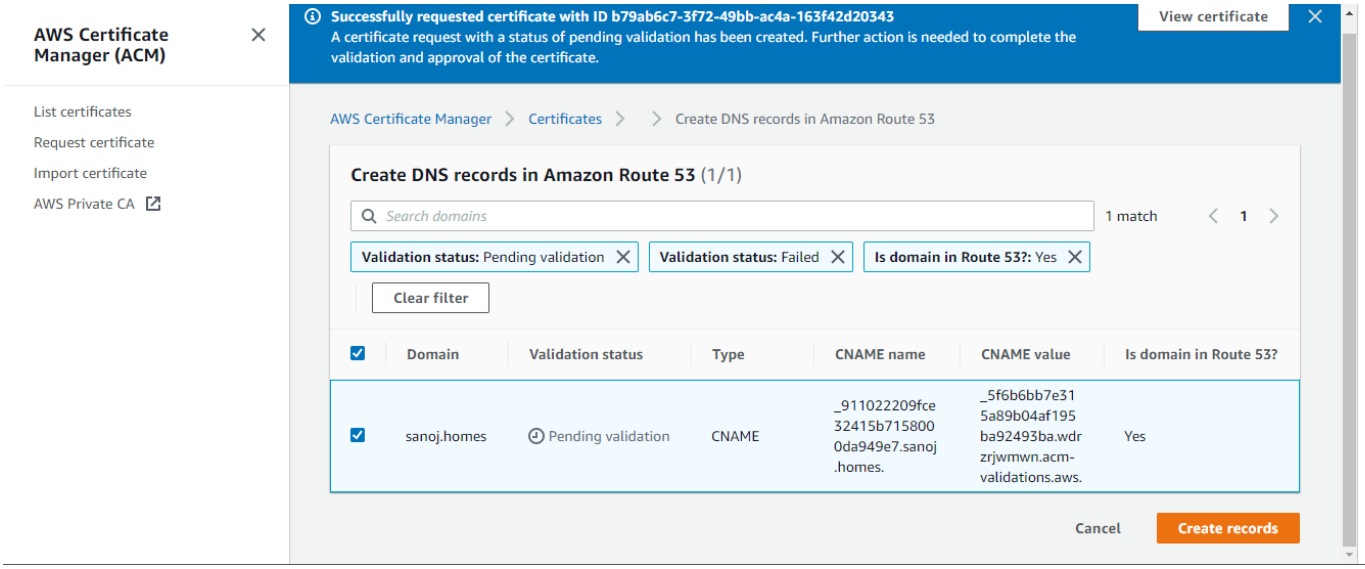
Step 5: Now click on **"Certificate ID."** after clicking on the certificate ID you will see below kind of interface.



Step 6: Now click on the **"Create Records in Route 53"**

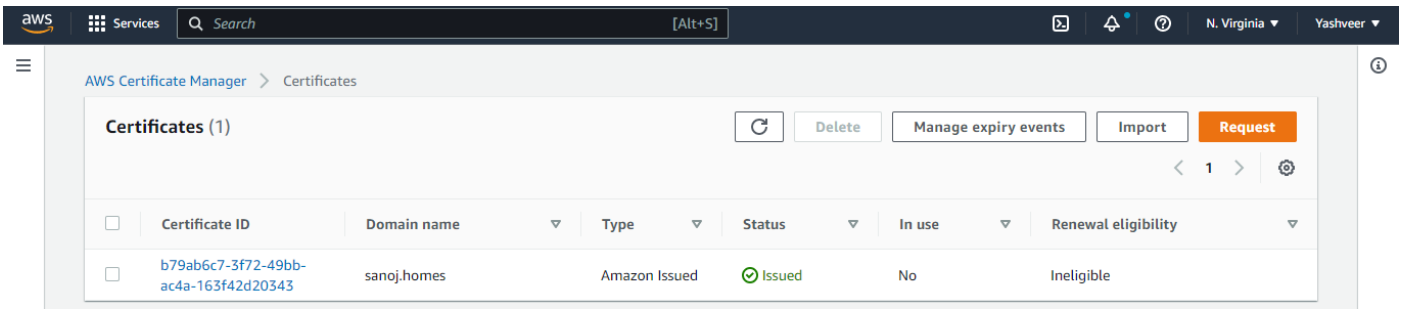


After that you have to click on the "Create Records" it will create a record on the route 53.



After clicking it will create the record in your route 53 automatically or you can also create a record and put the CNAME value in the record choice is yours. right now I have created from here. after that it will take upto 30 minutes or sometime it will take upto 48 hours it depends in my case it is issued within the 17 minutes. so wait till it shows issued.

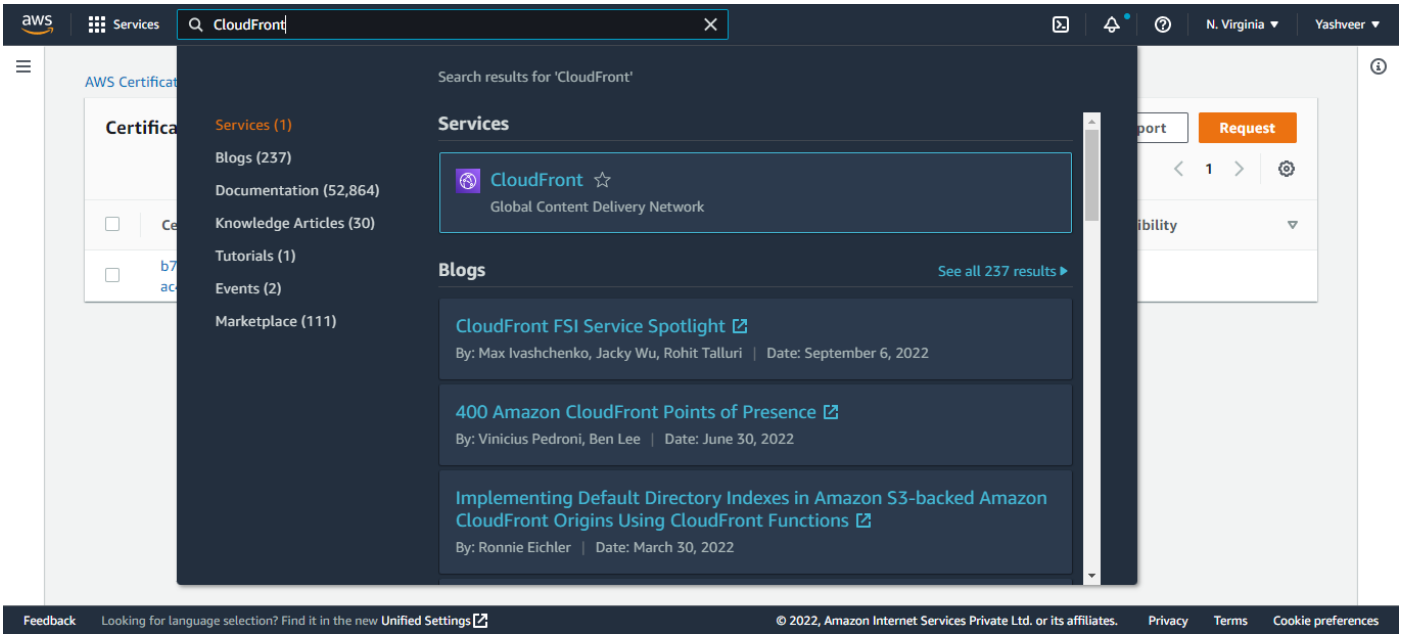
As you can in the below image my certificate is issued successfully.

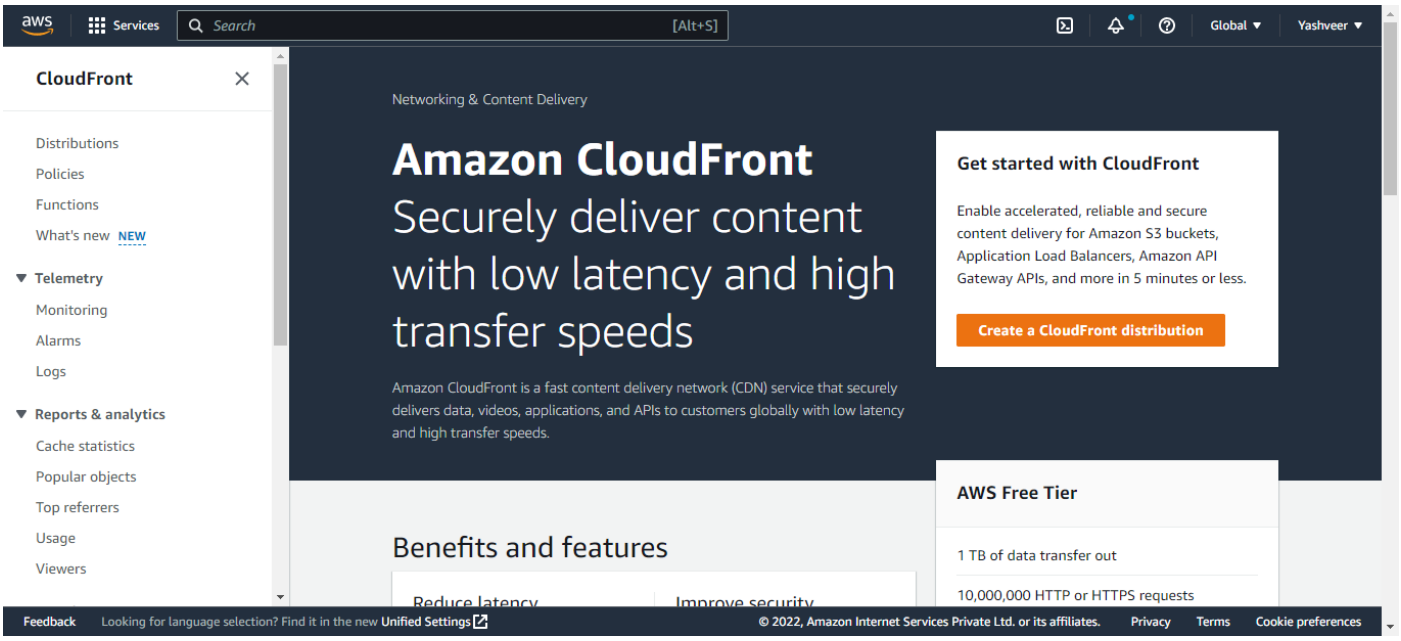


Now we are done with the ACM(Amazon Certificate Manager) Part.

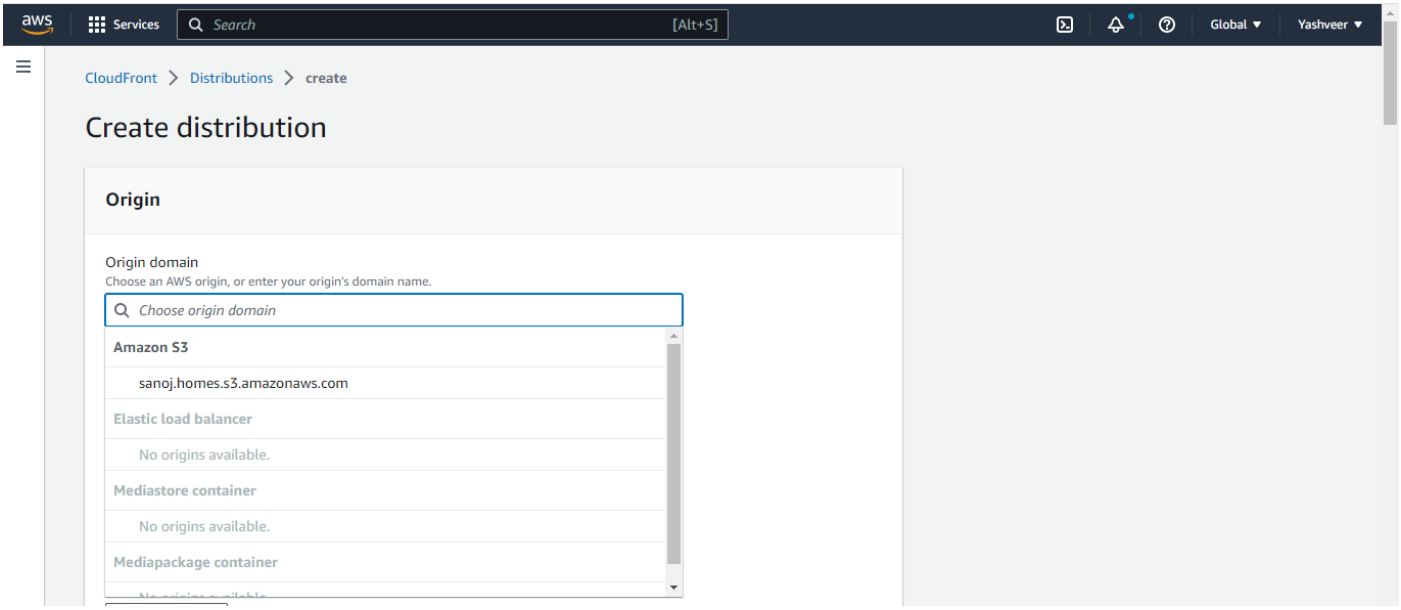
Setting up CloudFront:

Step 1: Search the cloudfront from the AWS console search bar and hit enter.

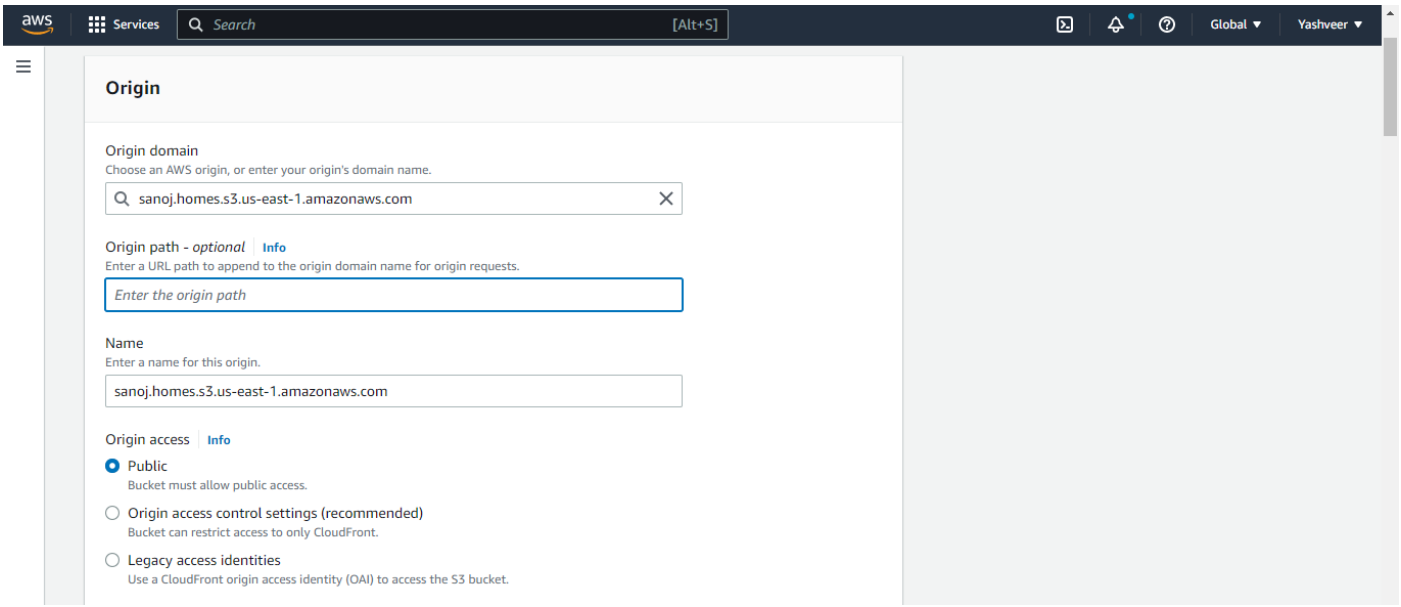


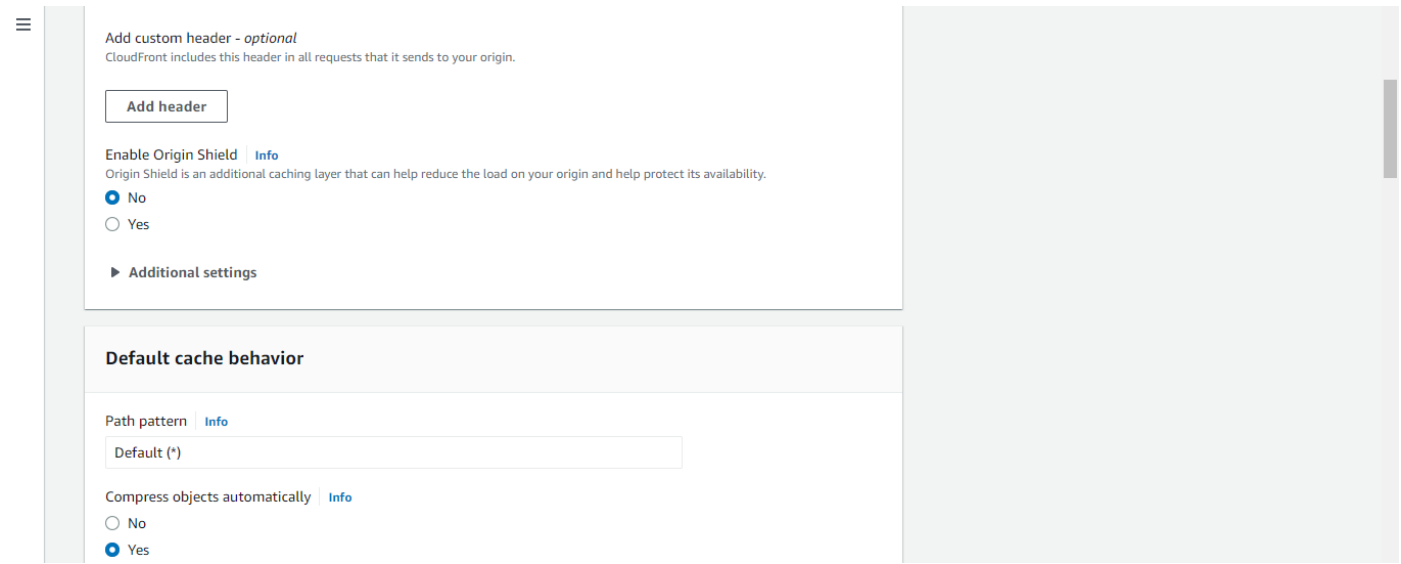


Now click on the "Create CloudFront distribution" after clicking on the Create cloudfront distribution you will see below kind of interface.

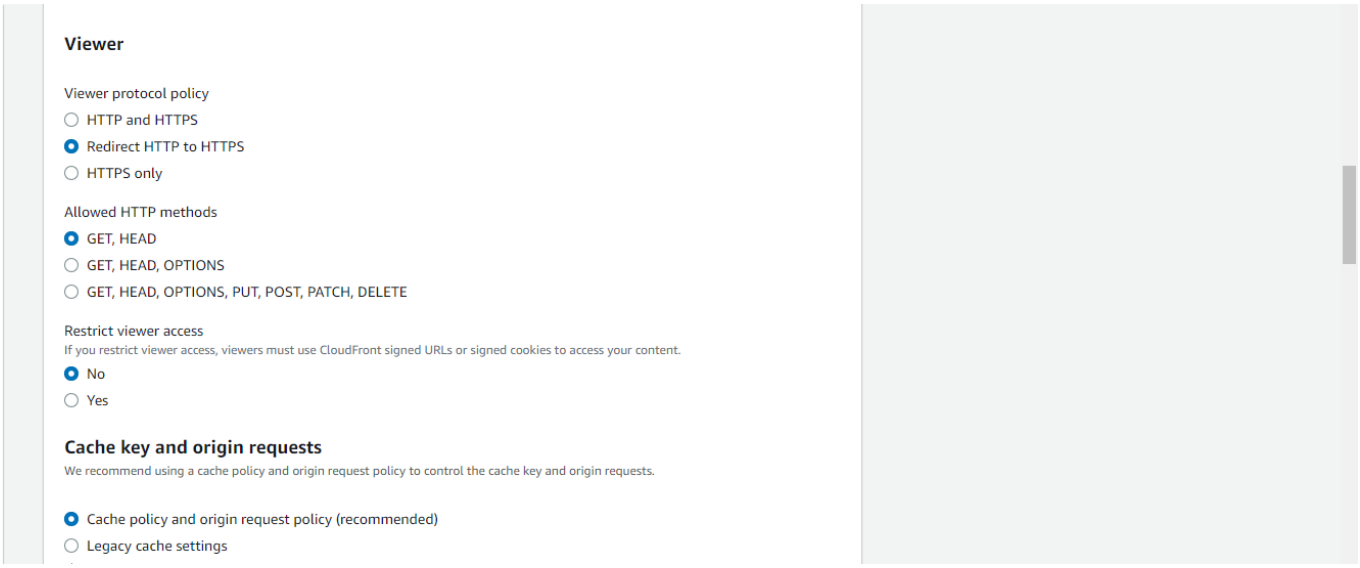


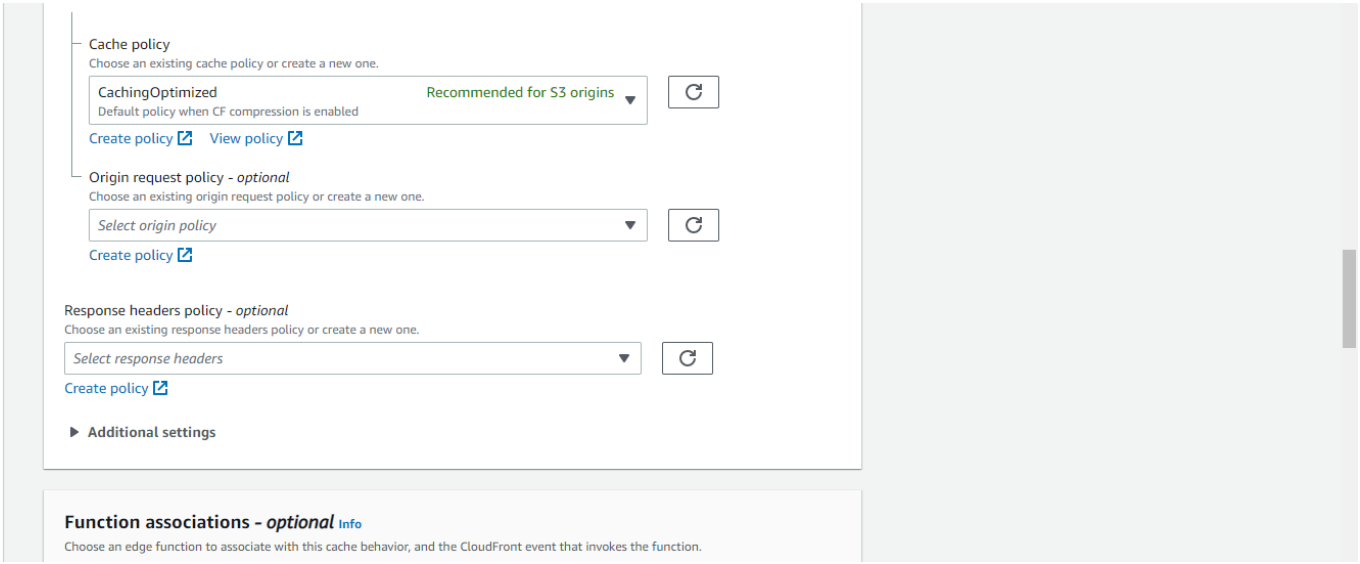
Select the Origin domain it will reflect automatically when you click on the “Choose origin domain” Search bar. so select here S3 origin domain because our website is hosted on the S3 domain.

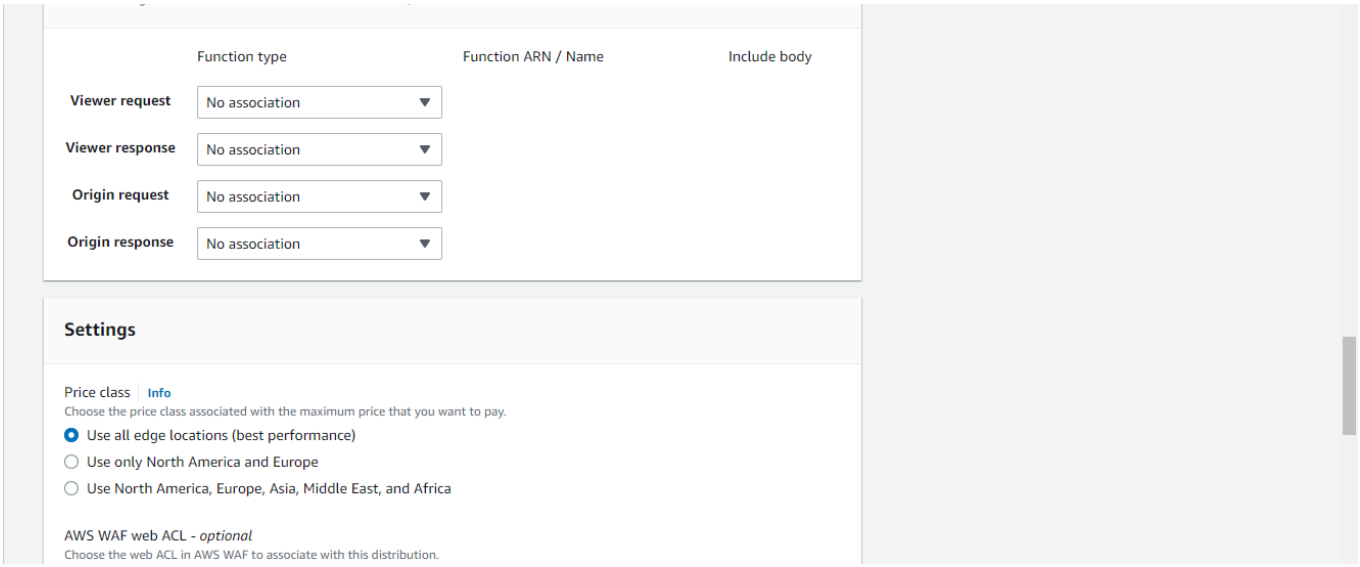




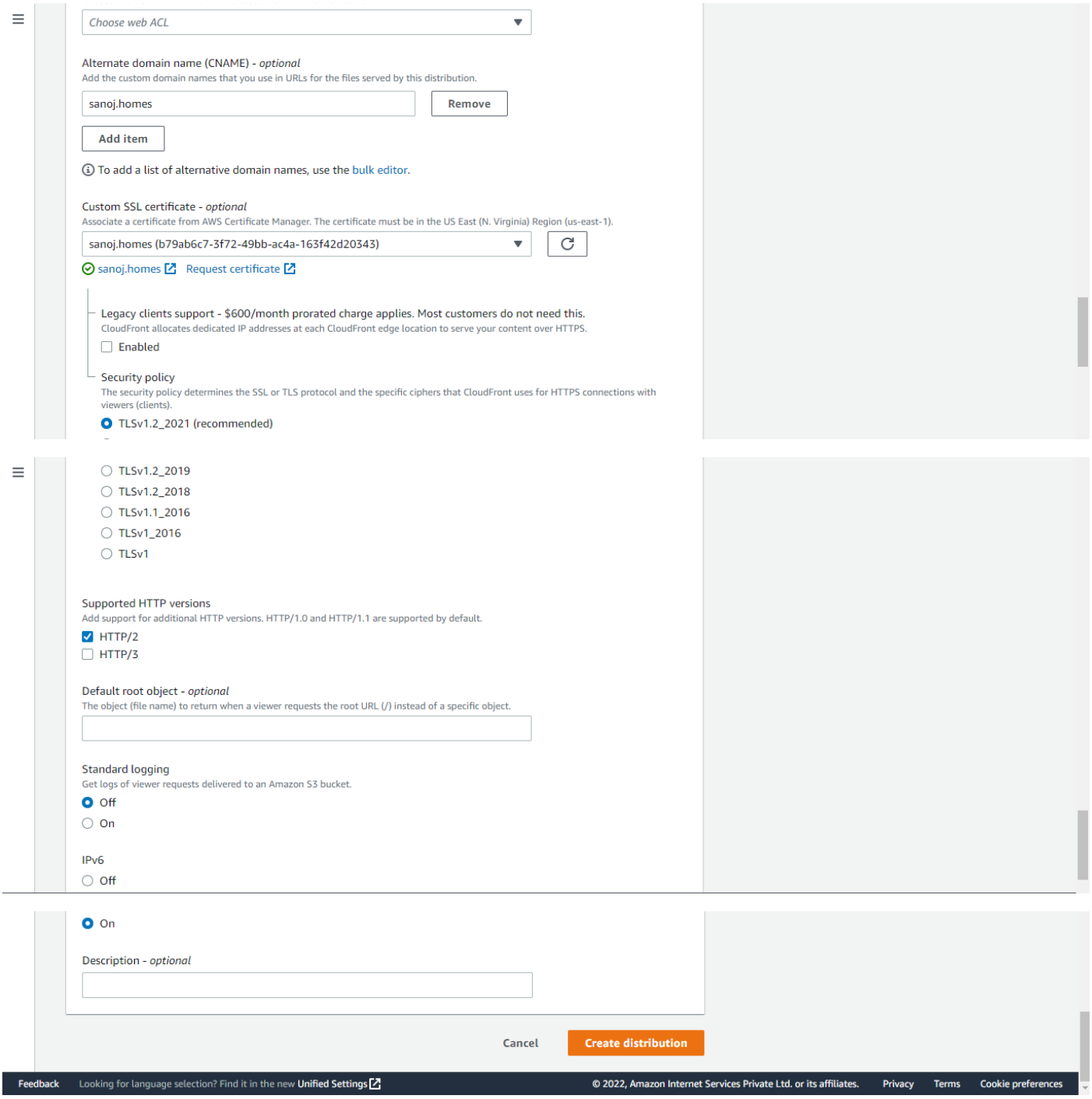
Select Viewer **“Redirect HTTP to HTTPS”**





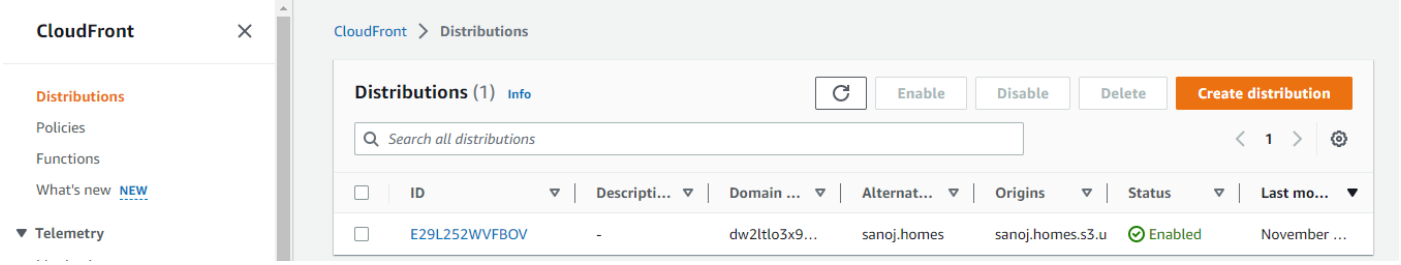


Now click on the **“Add item”** and put there your domain name and click on the **“Custom SSL Certificate”** to add the SSL certificate as you can see in the below image.

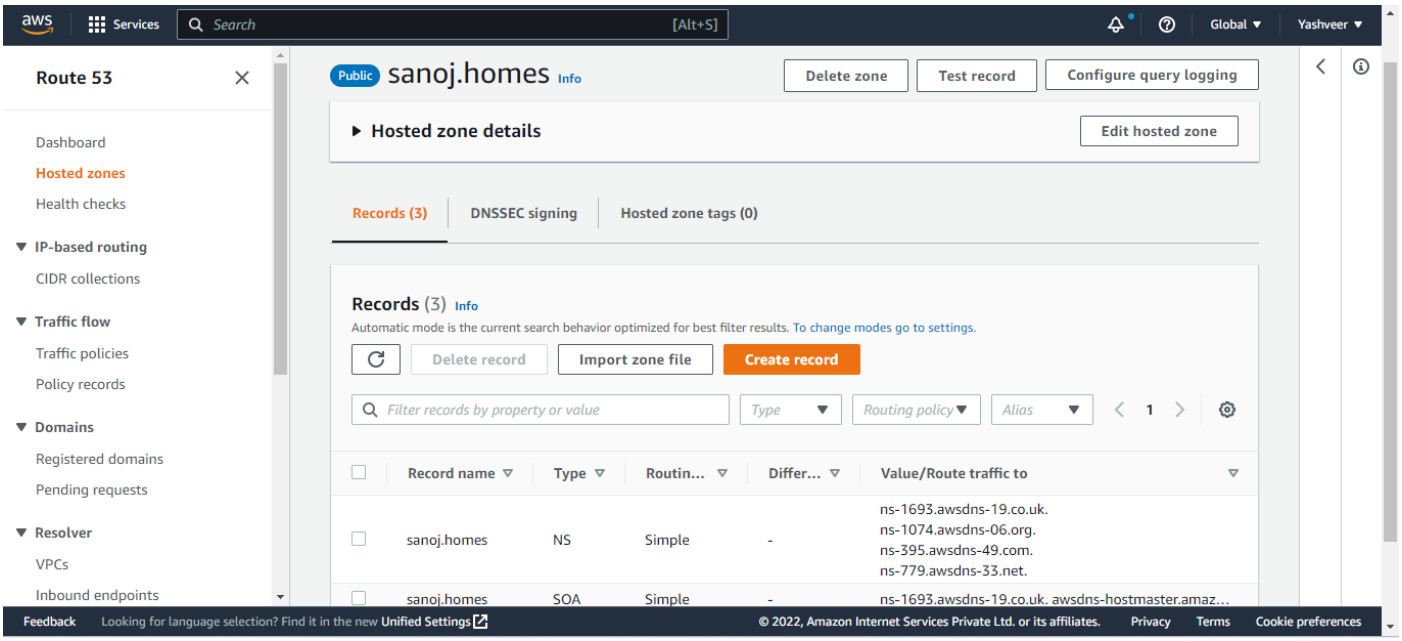


Now click on the **“Create Distribution”**

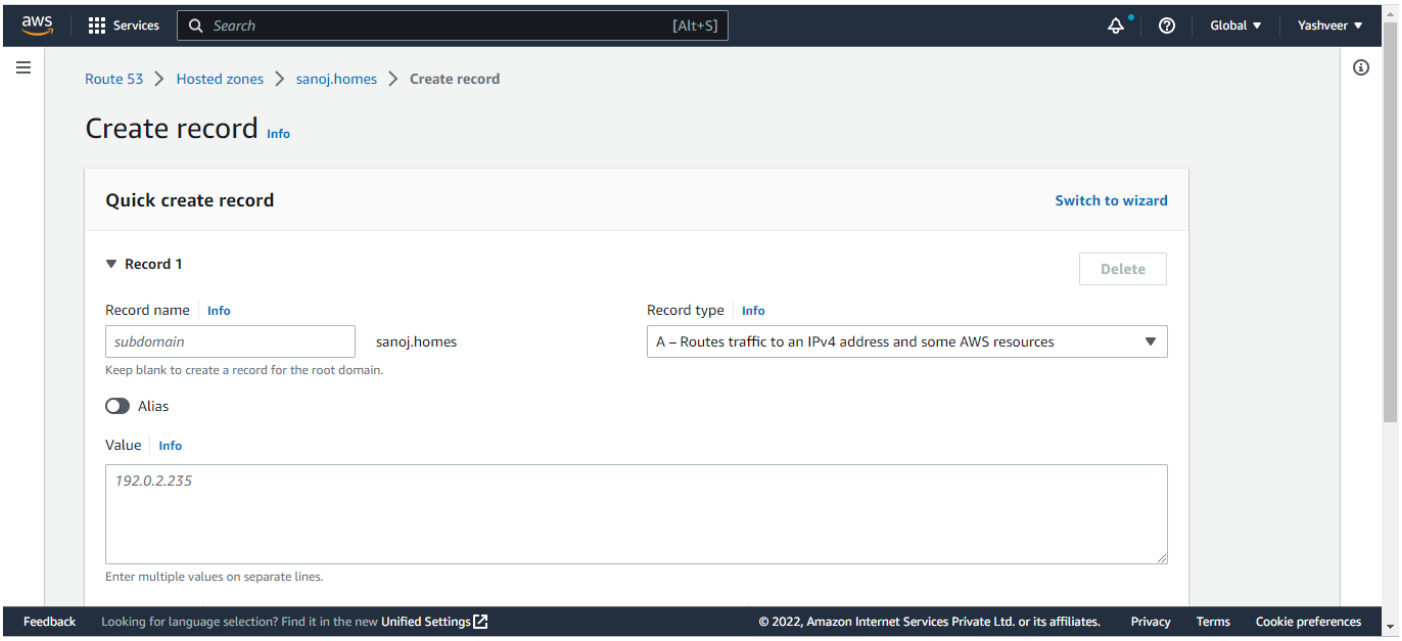
After clicking on the “Create Distribution” it will create the distribution and it will take few minutes approx. 10 minutes after that it will available to use and you will see below interface.



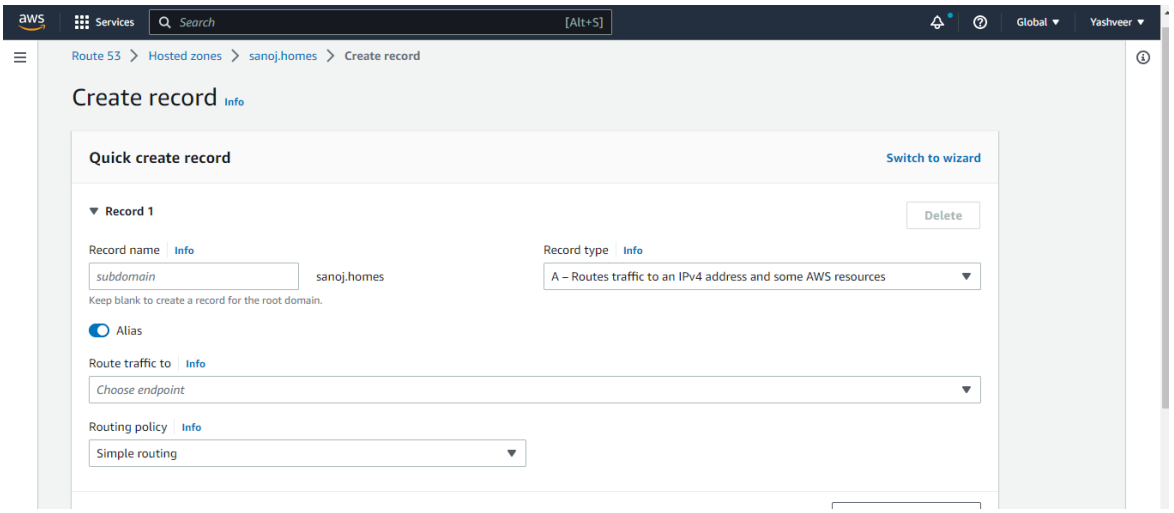
Now Again go to the Route 53 and Create a Record that point to the route to the CloudFront.



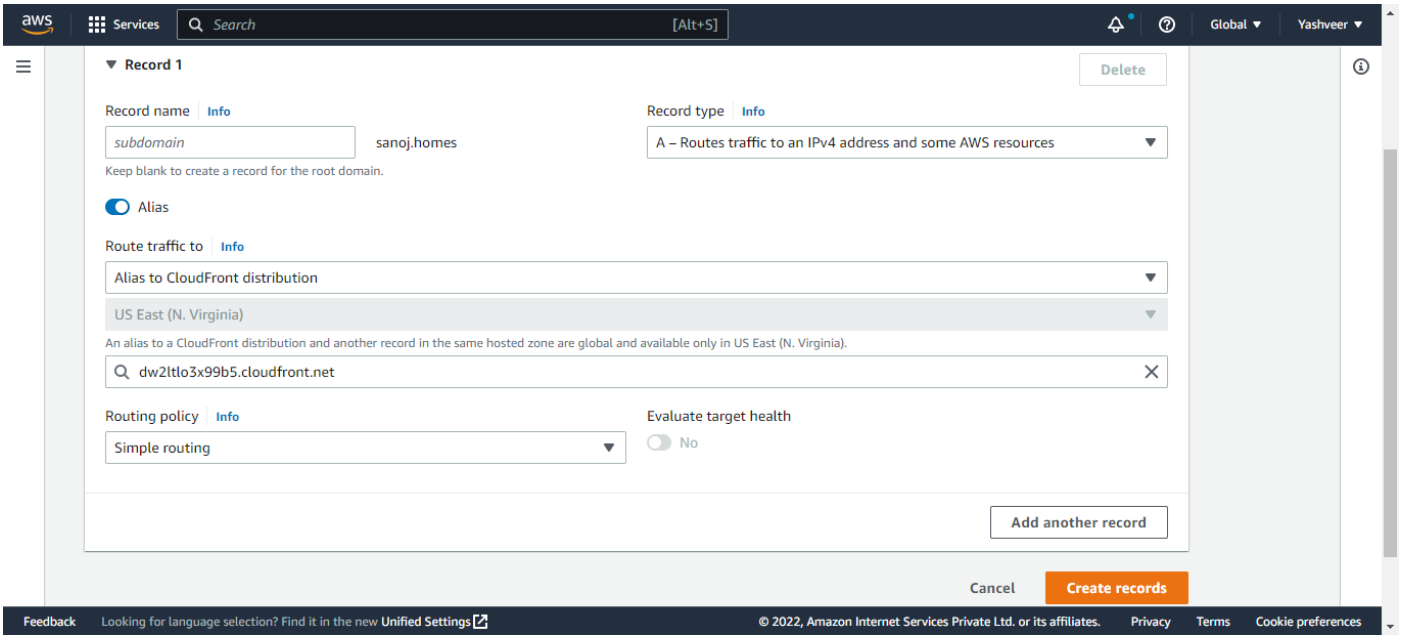
Now click on the **"Create record"** after click on the create record it will give below interface.



Click on the "Alias" button.



Now you have to click on the Route Traffic Search bar section and “search your cloudfront or you can select. then choose distribution and click on **“Create record”**



Now Hit your URL it will work in my case URL is sanoj.homes