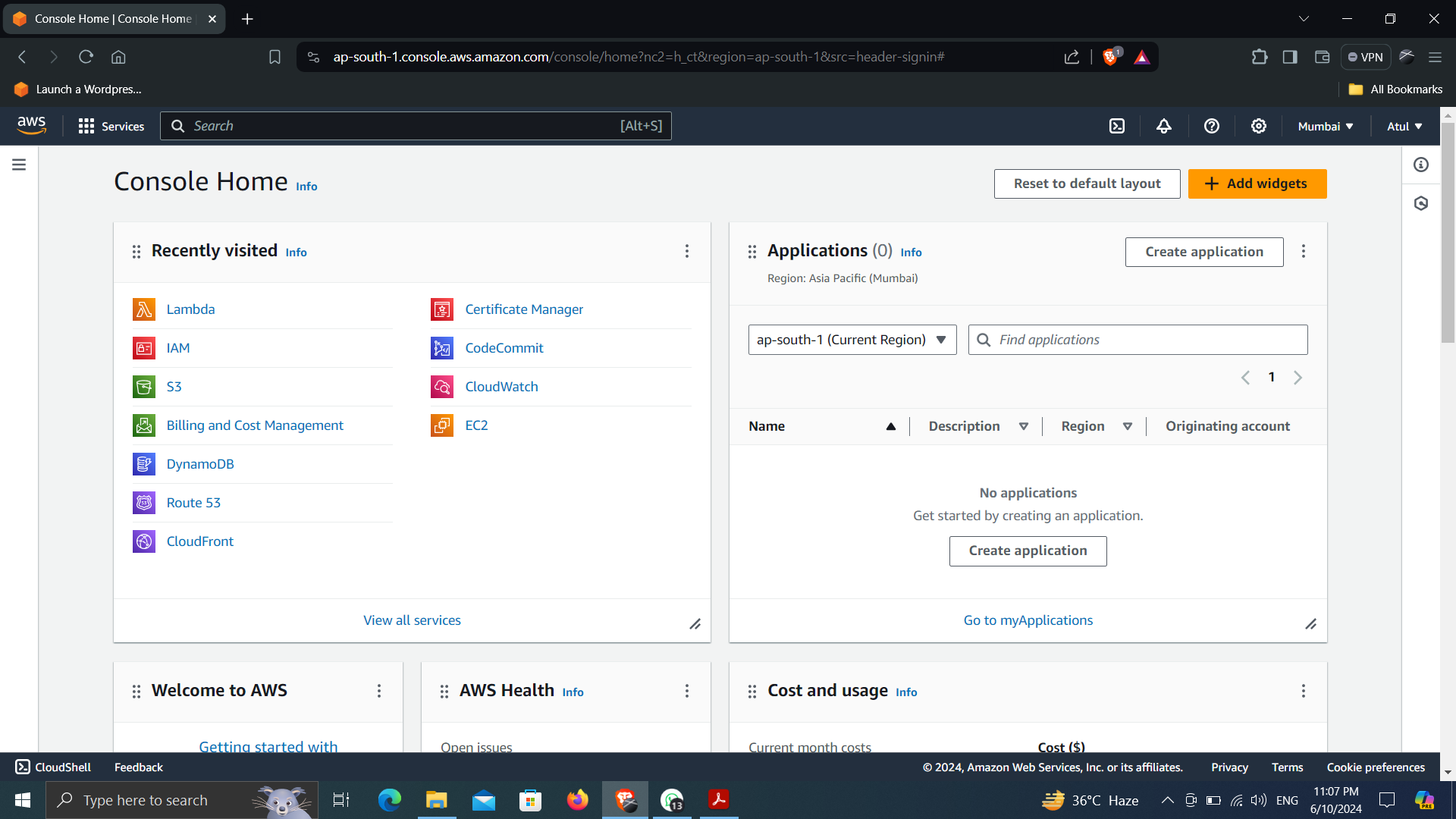
**Serverless-web-app-on-AWS**

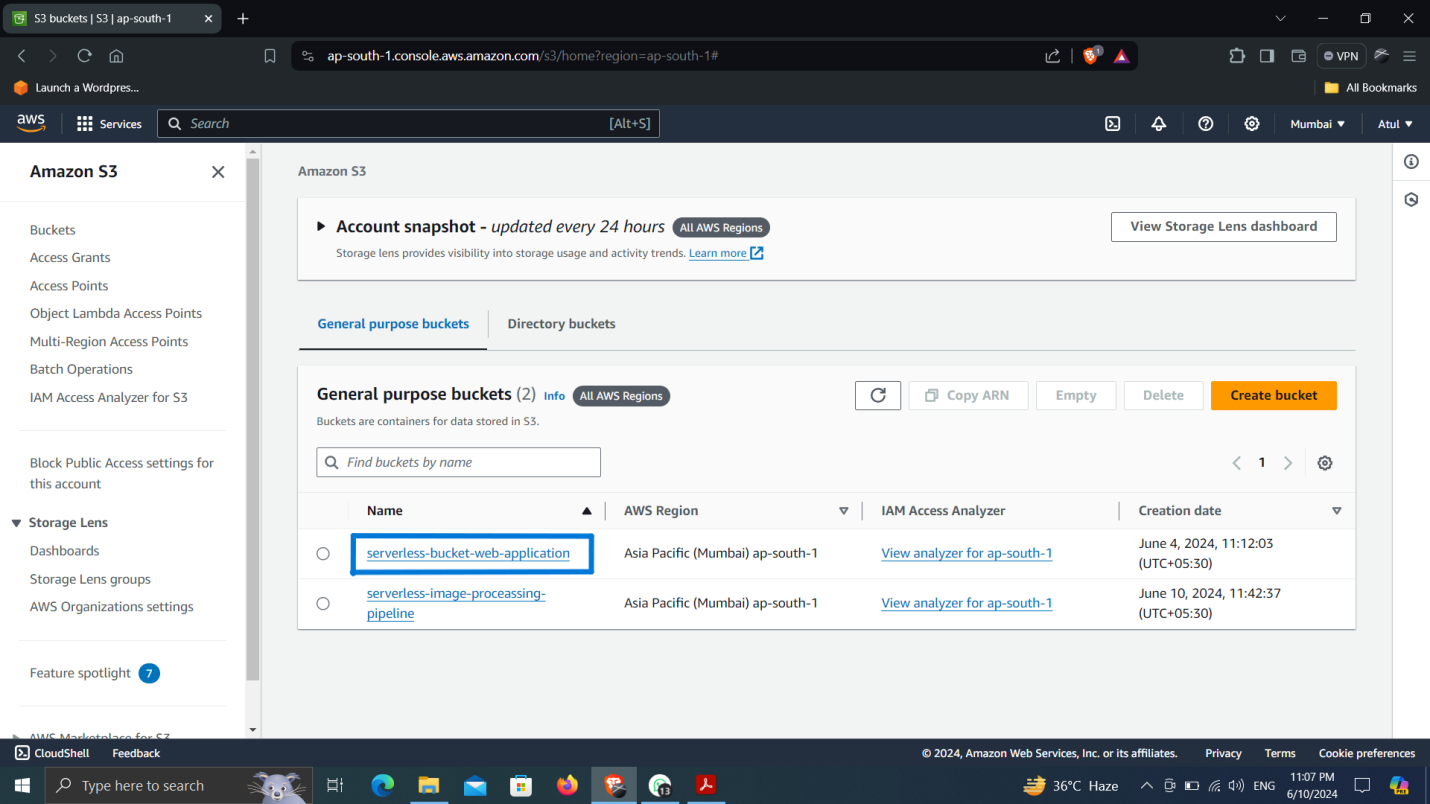
**Steps to build the Project:**

**Step 1: S3 Bucket**

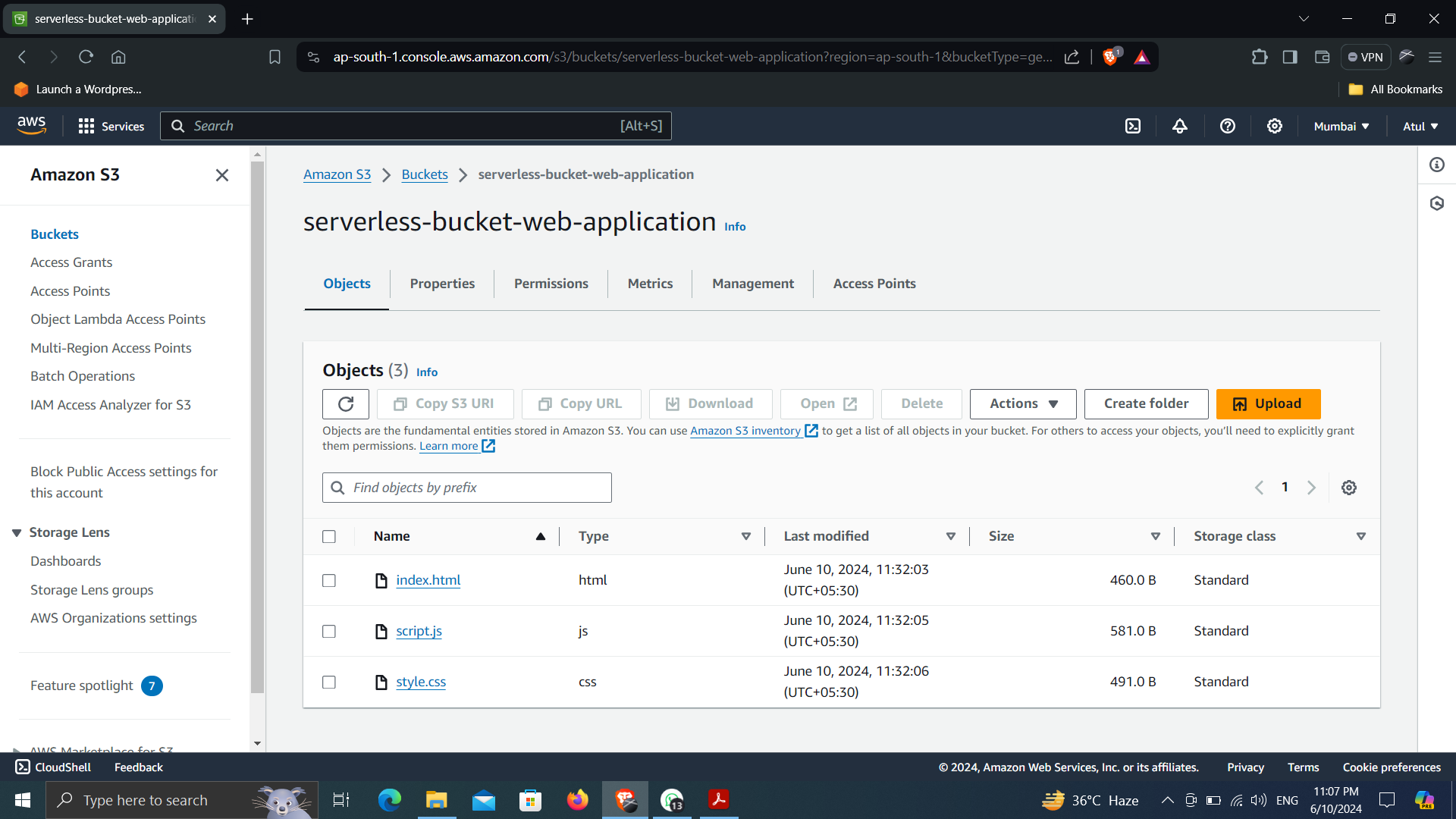
1. **Login into your AWS account using Login Id and Password.**

****

1. **Search for S3 Bucket**
2. **Create S3 bucket. Give a name to the bucket as “serverless-bucket-web-application” and set the region as “ap-south-1”**
3. **Click Create Bucket.**

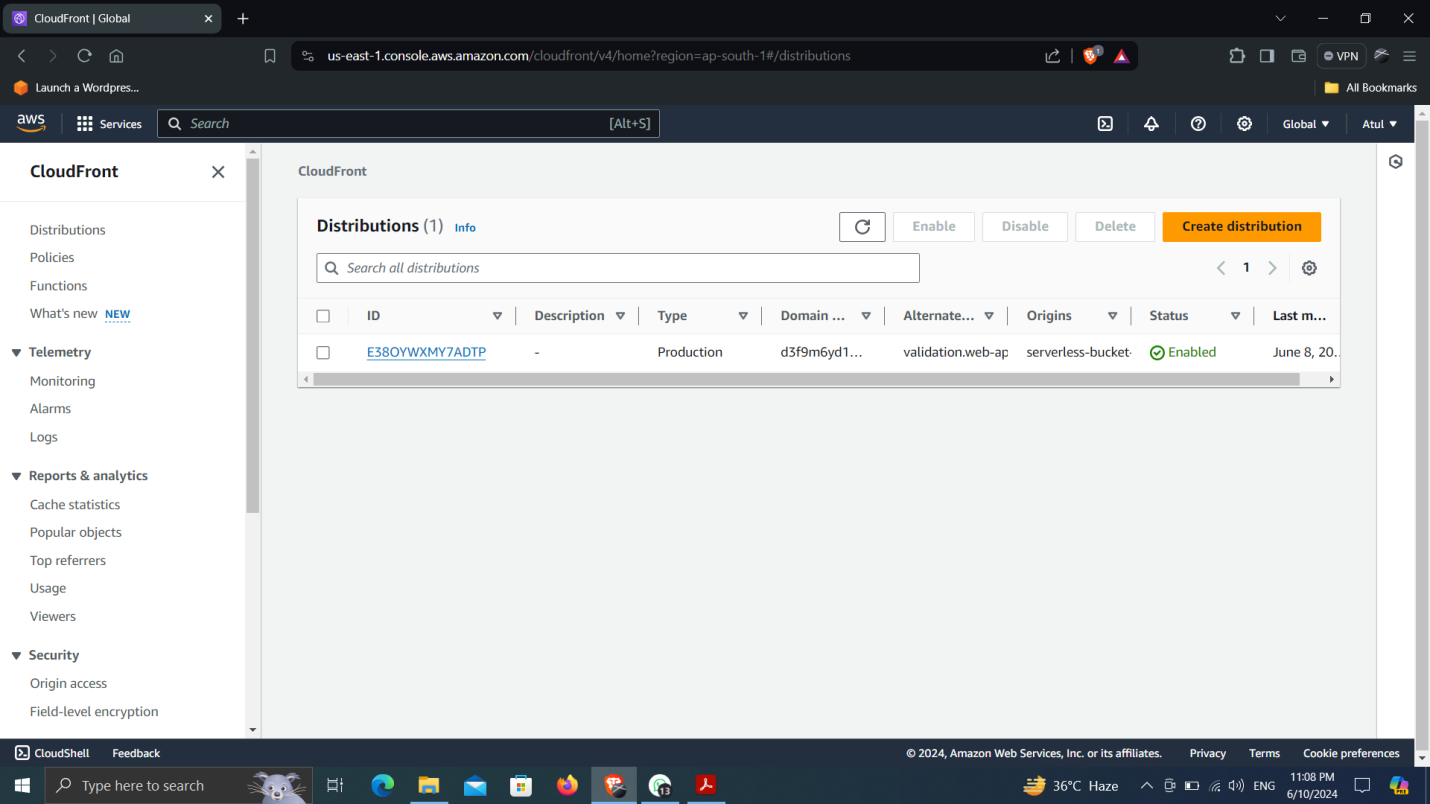
****

1. **Open the bucket, then click on upload and then upload the html, css & js file to the bucket.**

****

**Step 2: Create Amazon CloudFront**

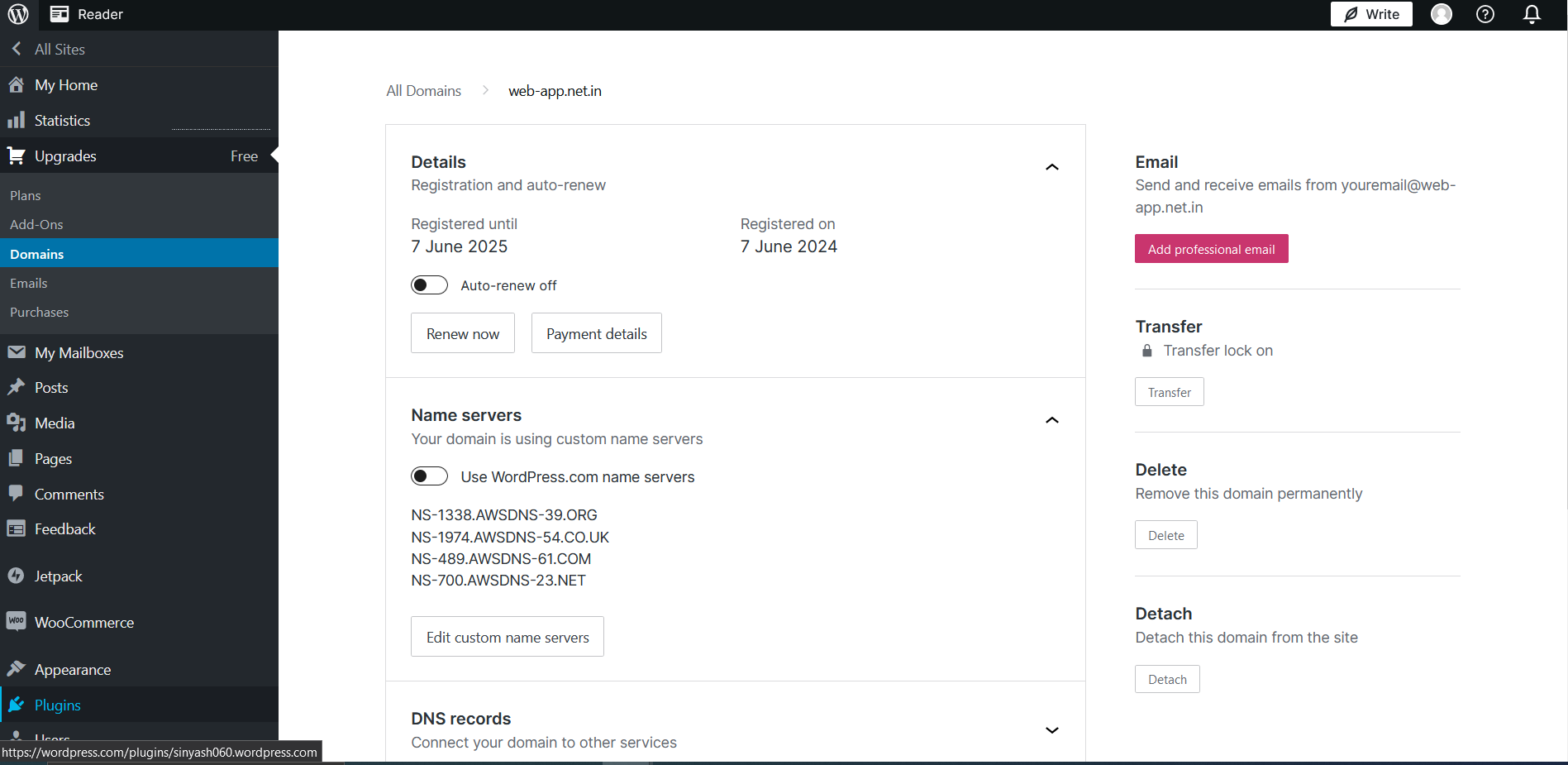
1. **Search for CloudFront in search bar in aws console.**
2. **Click Create CloudFront Distribution. Select Origin domain as “serverless-bucket-web-application.s3.amazonaws.com”.**
3. **Give name to the distribution as “serverless-bucket-web-application.s3.ap-south-1.amazonaws.com”**
4. **Change Origin Access to Origin access control settings which is recommended.**
5. **Click create control setting. Give a name to setting as “serverless-bucket-web-application.s3.ap-south-1.amazonaws.com-1” and keep Origin Type as S3 and then create.**
6. **Then click on create distribution. It will take some time to deploy the distribution.**

****

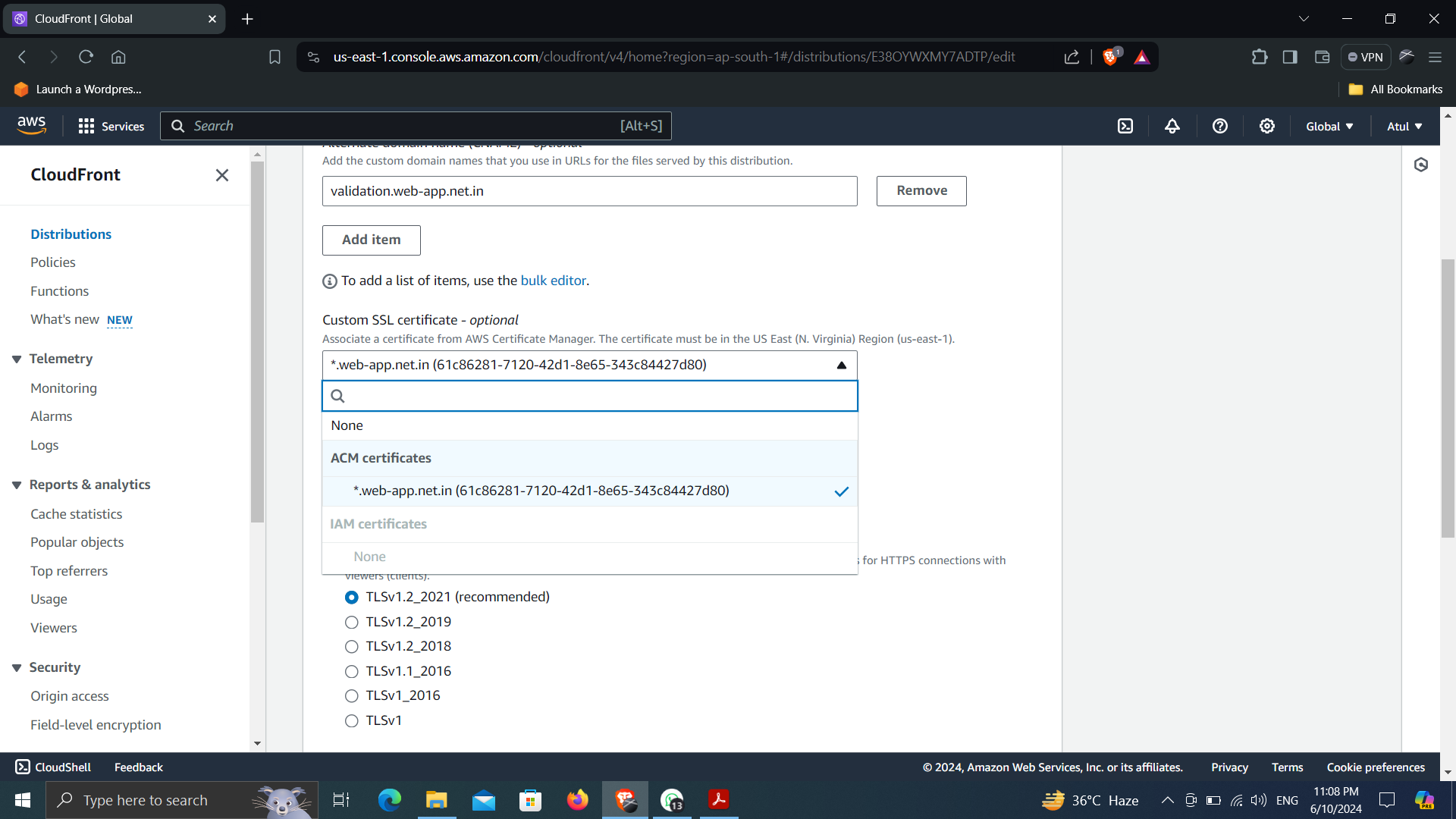
1. **Open the distribution. Then go to Origins then open the origin by selection the origin and then click edit. Then scroll down then click on Copy policy in blue box.**
2. **Then navigate to s3 bucket then open created bucket. Go to permissions tab, scroll down then you see the Bucket Policy box, select that and then click on edit and paste the copied policy and then save the changes.**
3. **Then come back to the cloudfront scroll down you see the settings, click on edit, then scroll down “Default root object - optional”, set it to “index.html” which is your html file name and which is the main entry point for your web application. Then save changes.**

**Step 3: Create Route 53 and ACM**

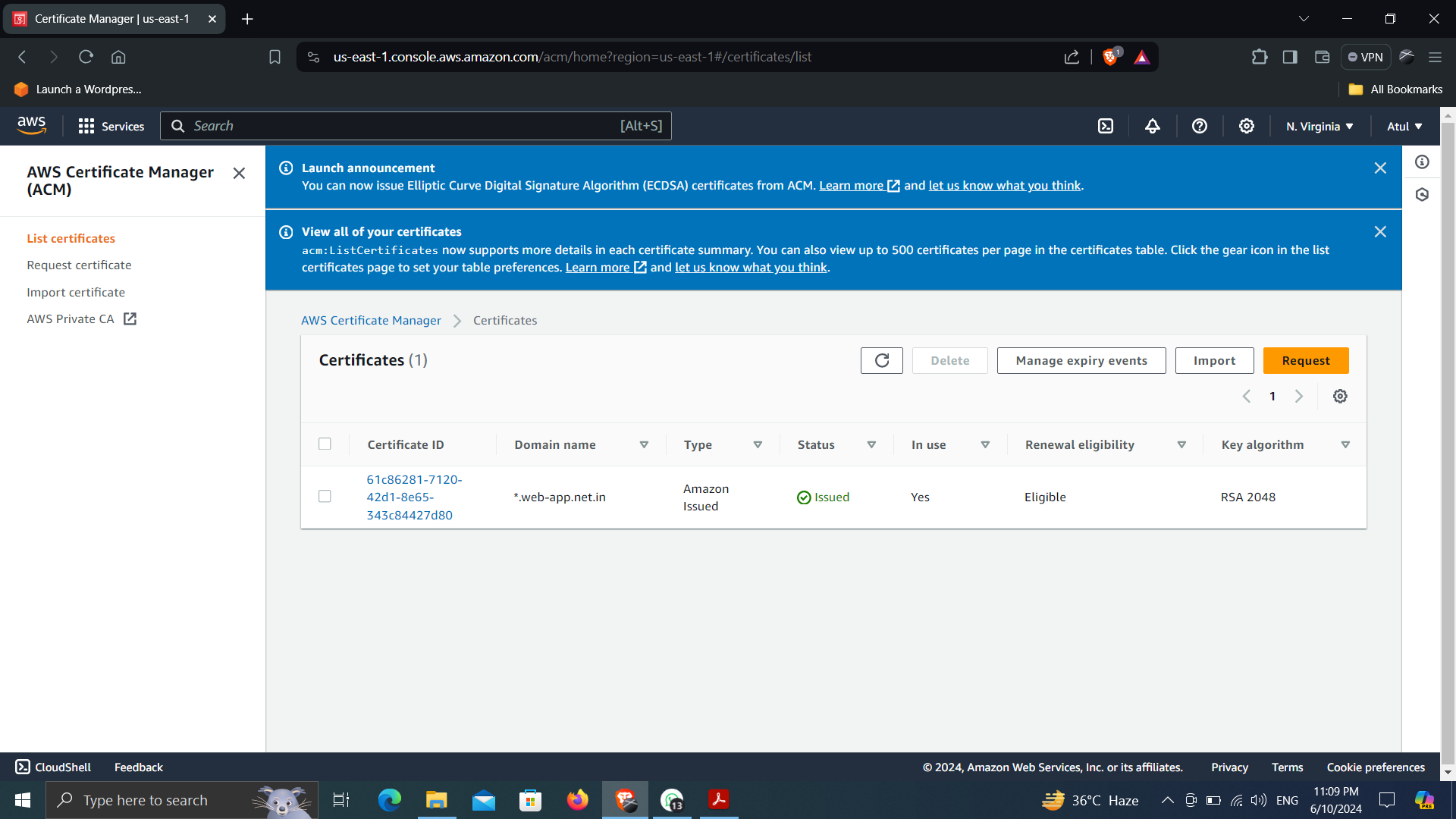
1. **Get a domain name, you can have free domain name from various domain providers like(freenom), I have taken name from WordPress.**
2. **In aws console search for Route 53, click on create hosted zone. Give name to the zone which is your domain name you get for free or you buy. Named as “web-app.net.in” then create.**
3. **Then you will get two records in hosted zone, one record have four nameserver named as “value/route traffic to”, copy them and edit in your domain from where you buy domain name.**

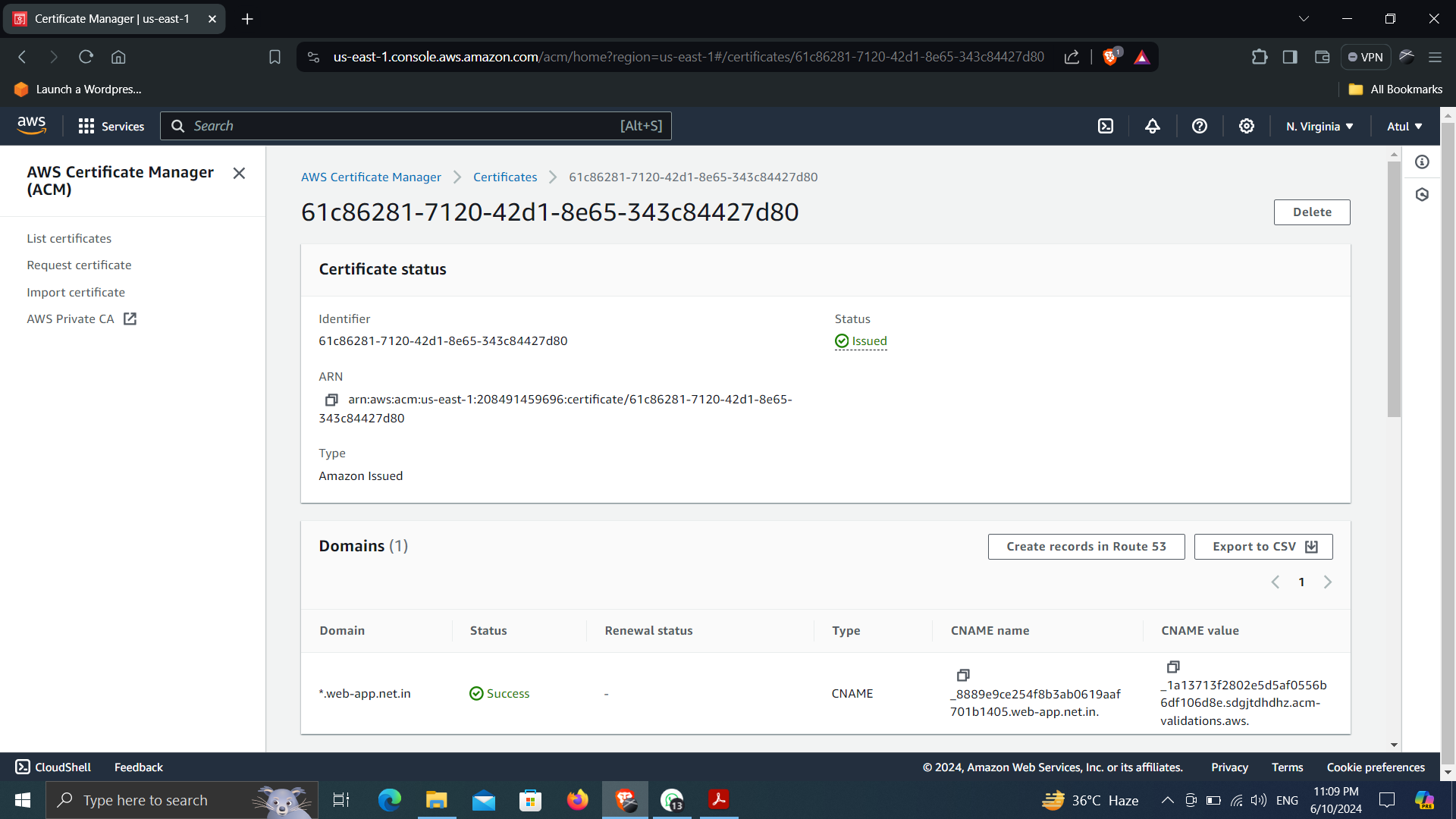
****

1. **Navigate to the CloudFront then settings, we have to add Alternate Domain Name, click edit on settings, give alternate domain name (ex. “name.domainname”) as “validation.web-app.net.in”.**
2. **Scroll down to the SSL Certificate, click on request certificate, then click to request a public certificate, then click next.**
3. **Give Fully Qualified Domain Name as “\*.web-app.net.in”. then click on Request. Certificate can take time from 10 minutes to 2 hours to get validate. Once certificate gets issued. Then go back to cloud front tab and select records in ssl certificate and save changes.**

****

1. **Open the certificate go to domains click on the Create Record in route 53. Then click create record.**

****

****

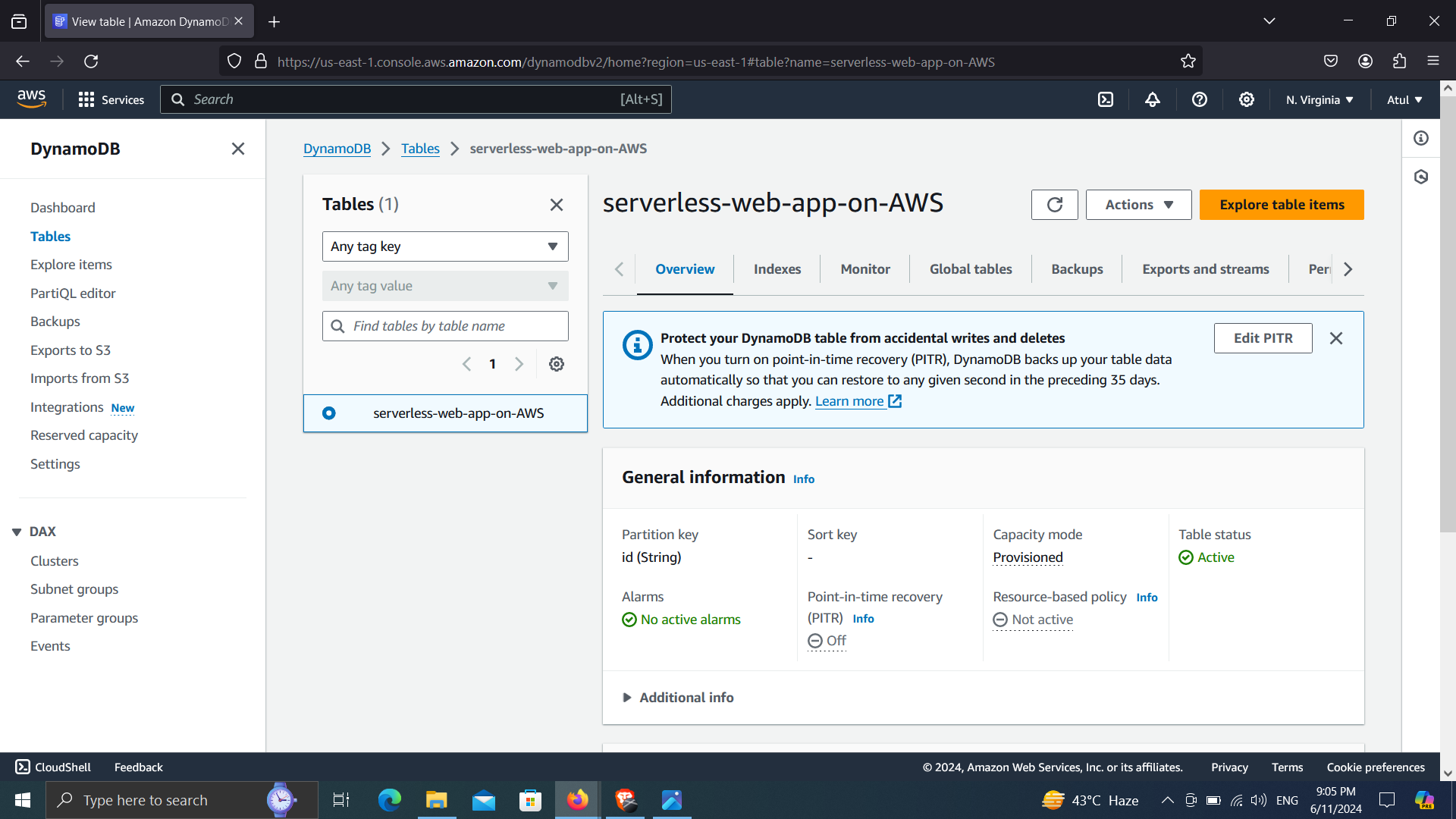
1. **Navigate back to route 53, you can see a new record created in the hosted zone, now create a new record from route 53, click on create record.**
2. **Give name to record name (subdomain name) as validation. Enable alias. In Route Traffic to select “Alias to CloudFront Distribution” and then choose distribution from created records named as “validation.web-app.net.in”. Click create records. Now check the website on browser if its running successfully or not.**

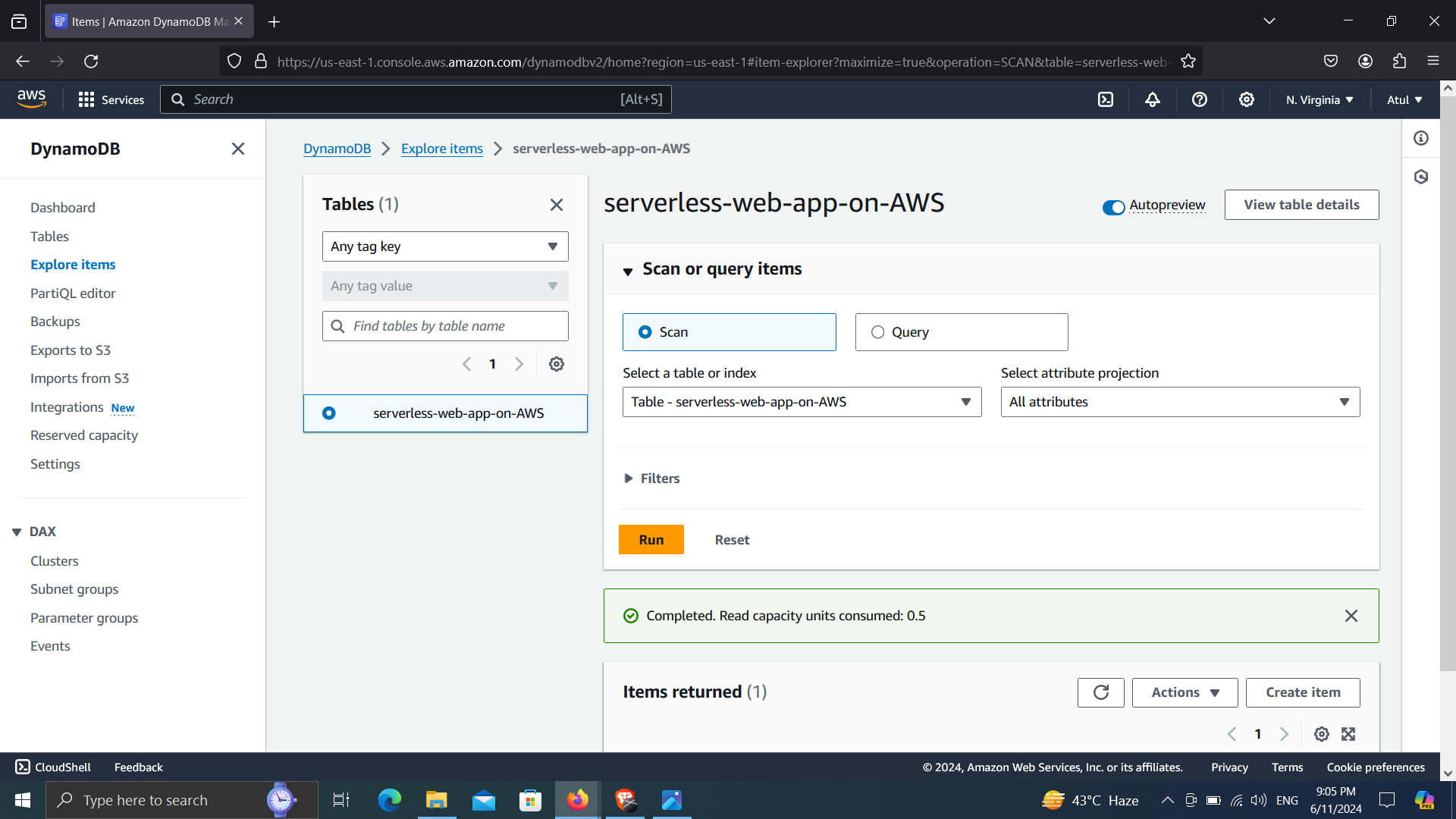
**Step 4: Create AWS DynamoDb**

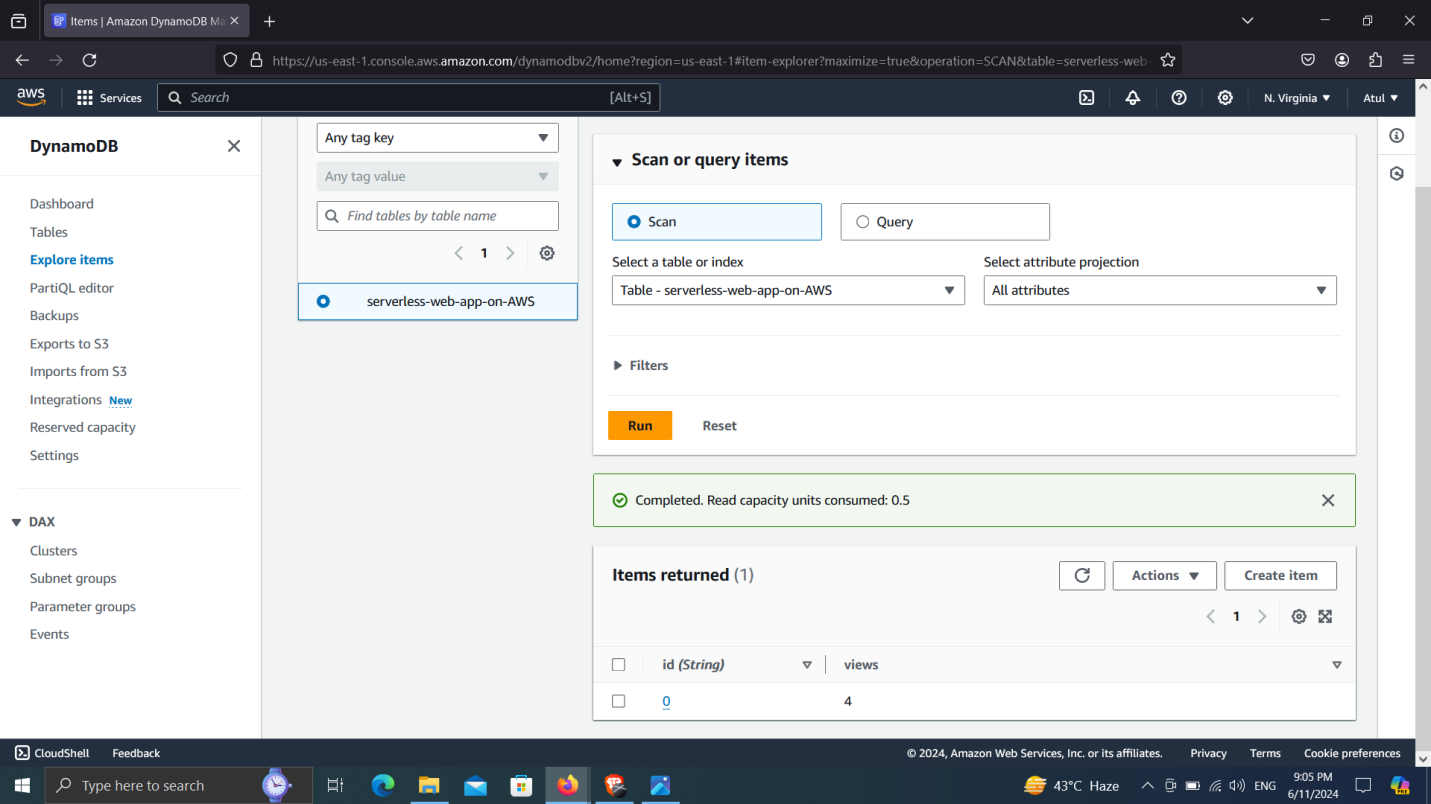
1. **Search for DynamoDb in AWS console, click on create table, give table name “serverless-web-app-on-aws”. Add “Id” in partition key and then create table.**

****

1. **Open created table, go to Explore table items, scroll down to item returned, click create item. Add value of attribute name “id” as 0. Click add new attribute. Add name as “views” and value as “1”, and then create item.**

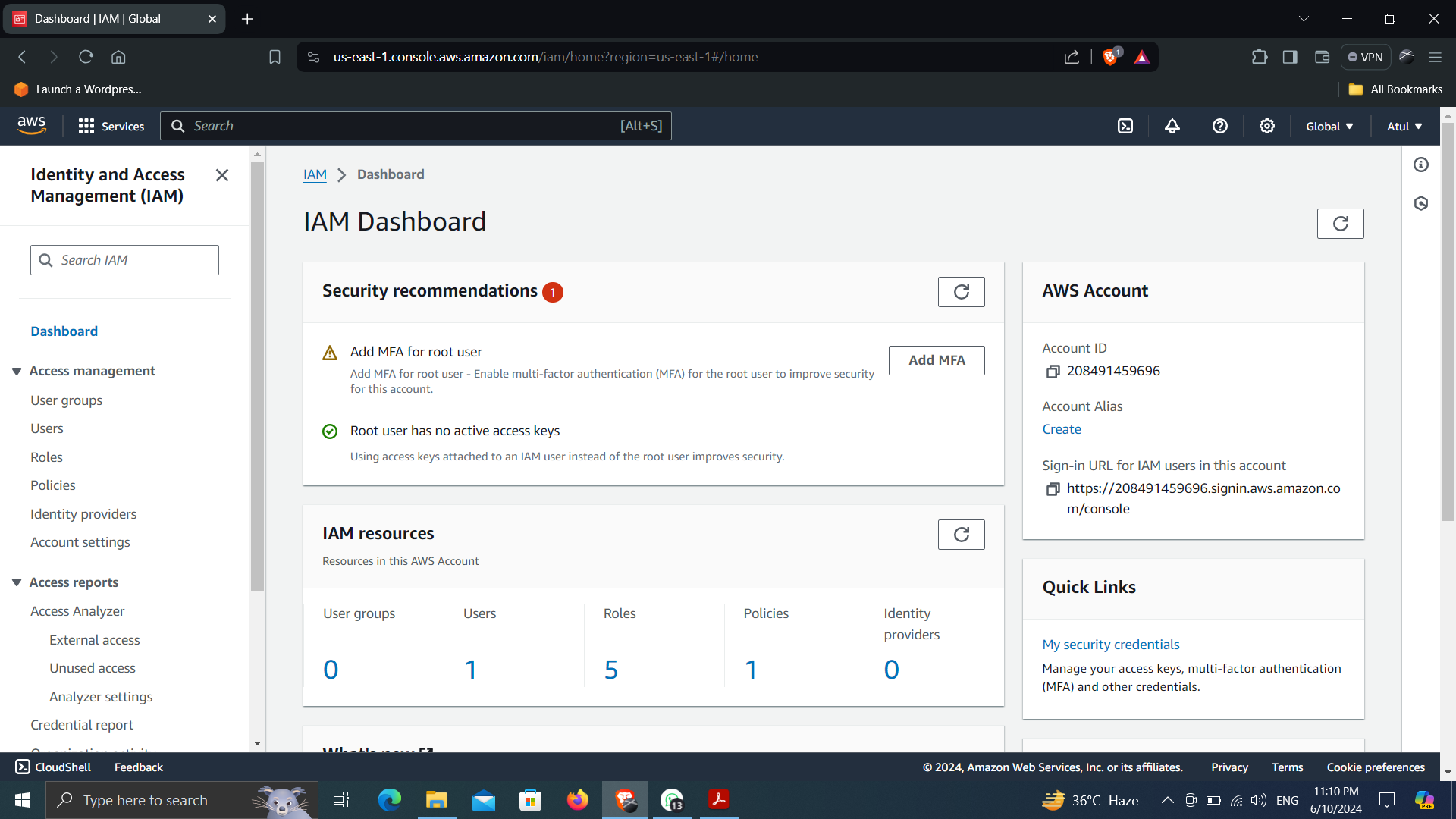
****

****

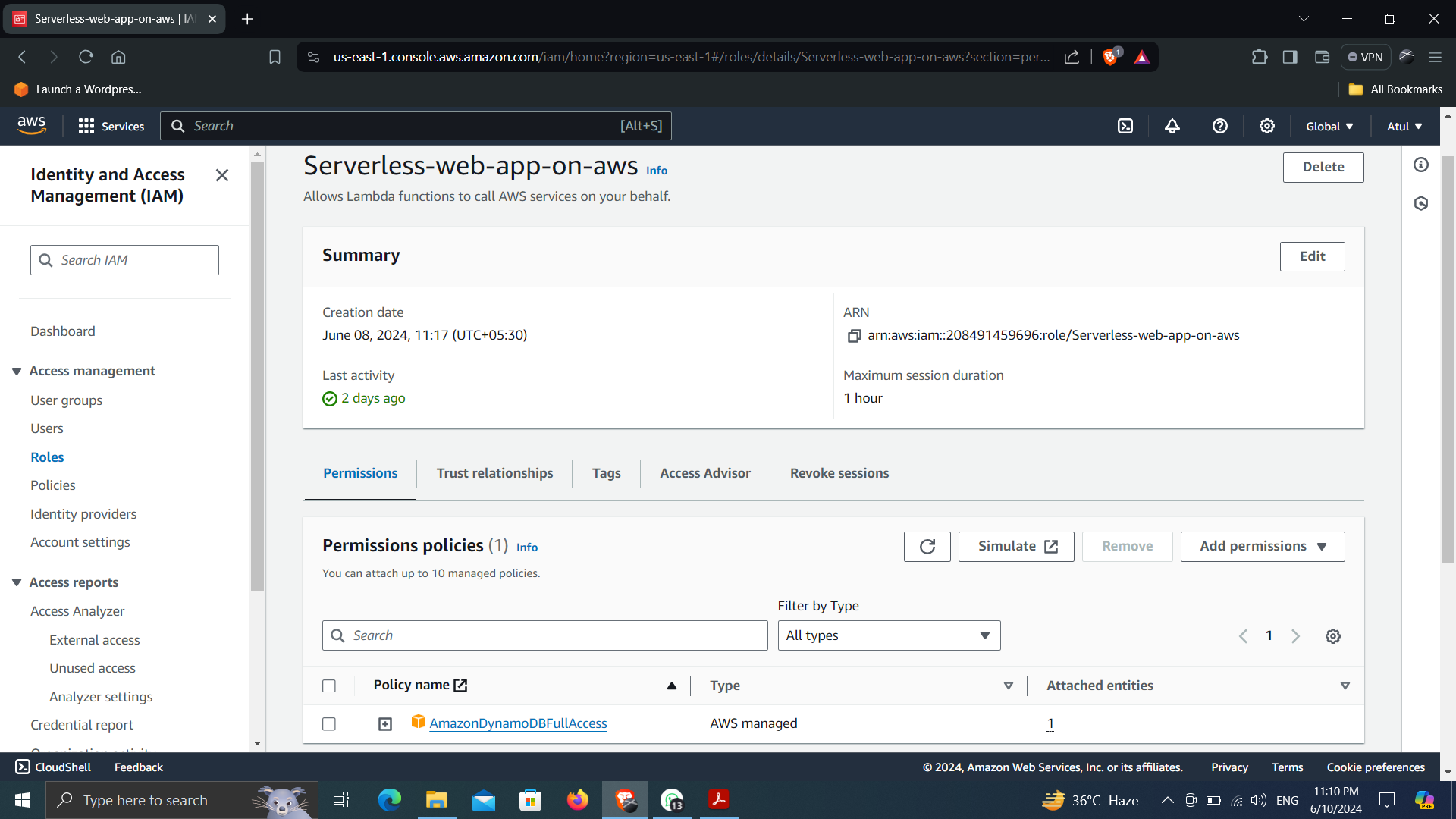
****

**Step 5: Create IAM role**

1. **Search for IAM in AWS console, in IAM go to Roles, click on create role.**

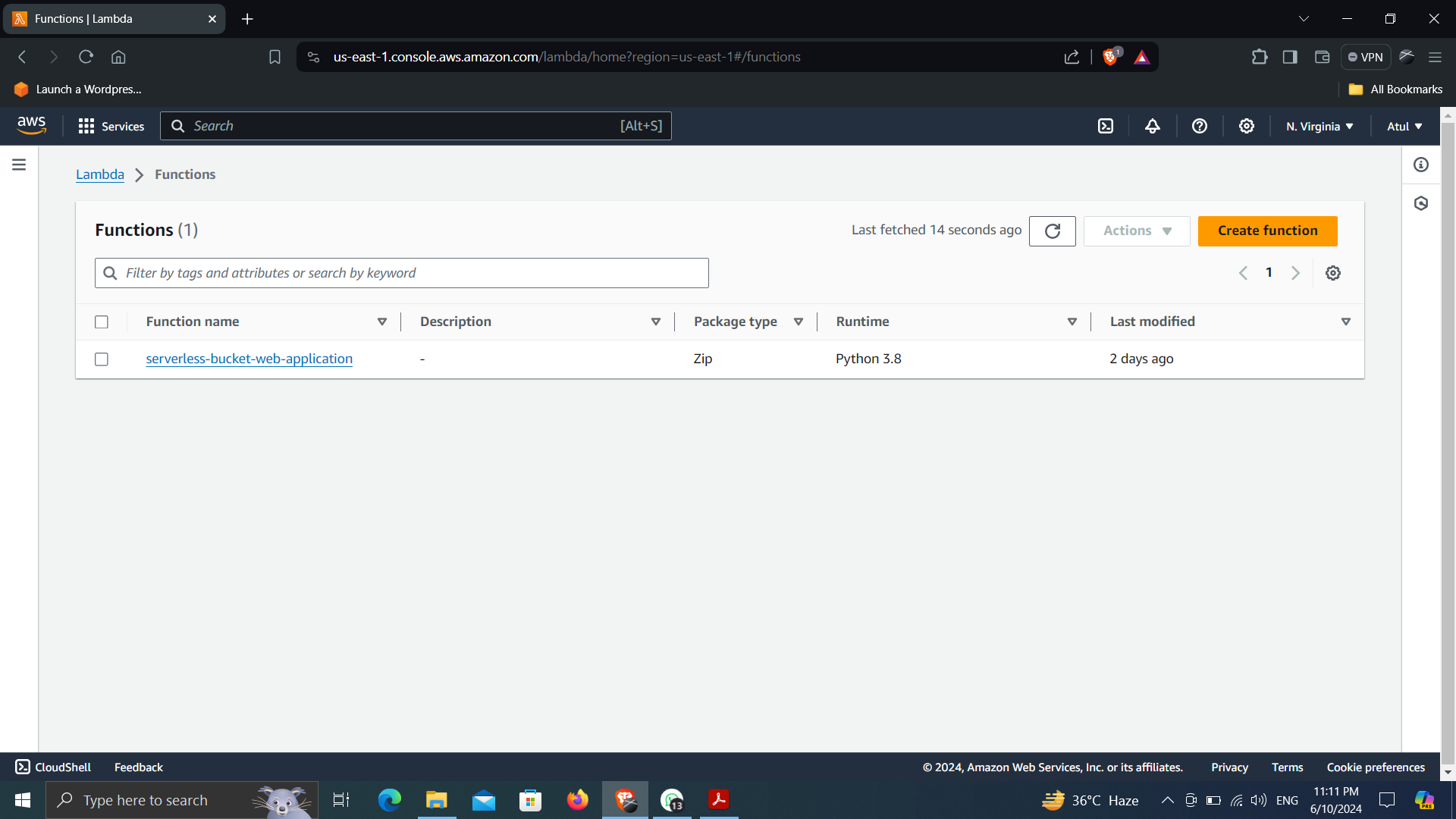
****

1. **Select Trusted Entity Type as “AWS Service”, then select Lambda in Use Case. Then Next.**
2. **Search for DynamoDB in Permission policies and then select “AmazonDynamoDBFullAccess” policy, then Next.**
3. **Give name to the role as “serverless-web-app-on-aws” then create role.**

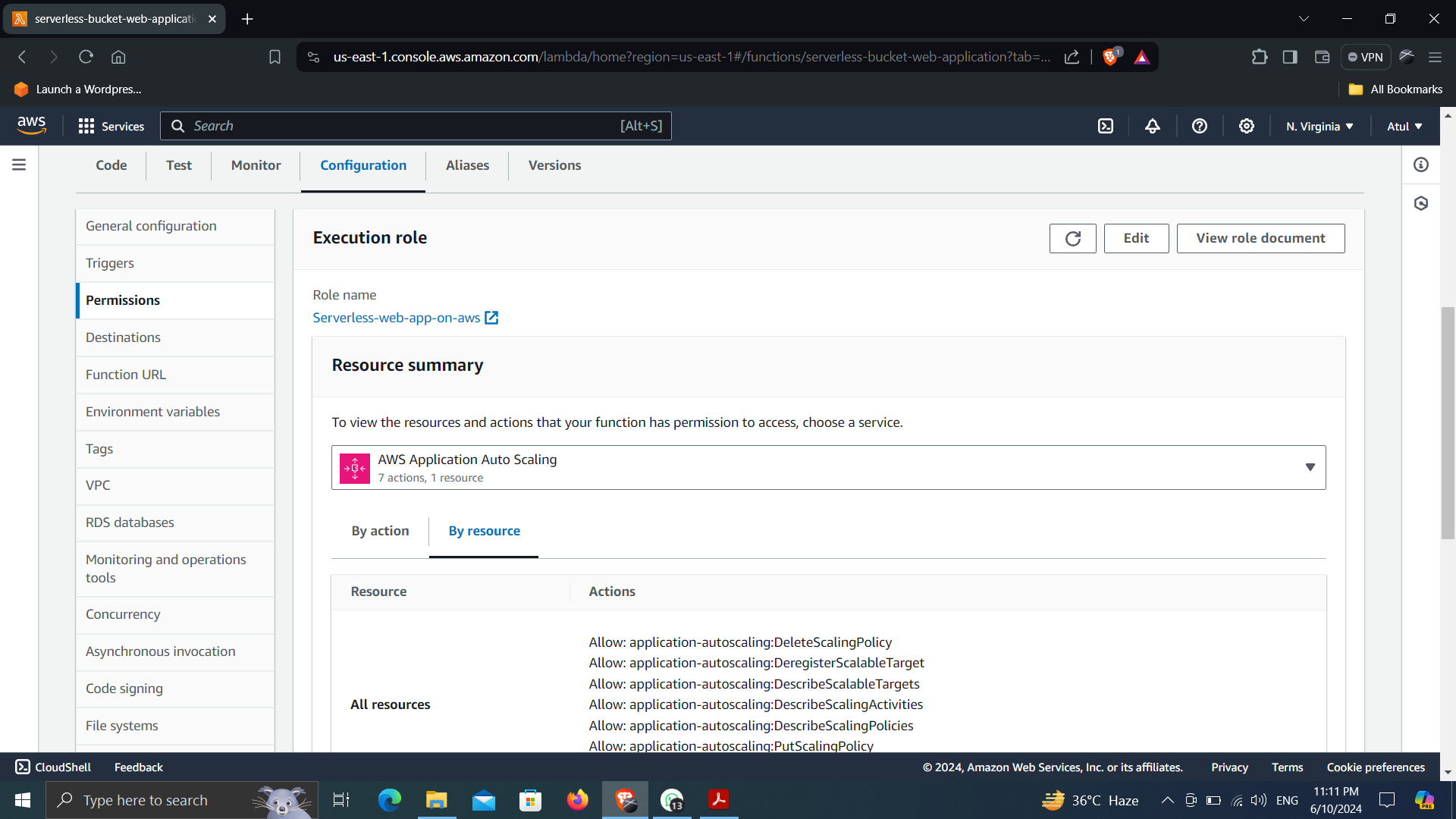
****

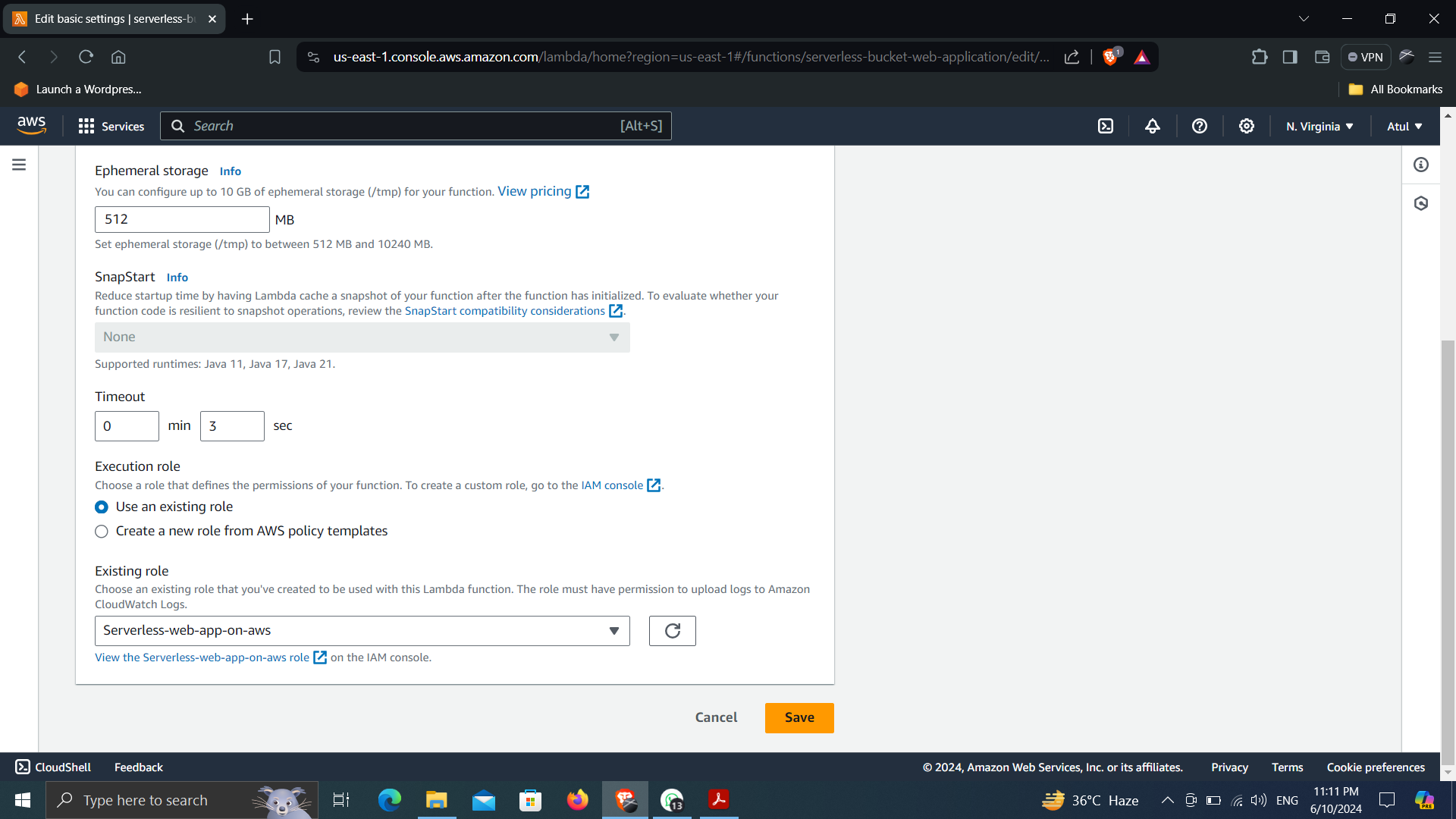
**Step 6: Create Lambda Function**

1. **Search for Lambda function in AWS console, click create a function.**
2. **Give function name as “serverless-web-app-on-aws” and choose runtime as “Python 3.8”.**
3. **Scroll down to Advance Setting, choose NONE in Auth type and choose Configure cross-origin resource sharing (CORS). Create function.**

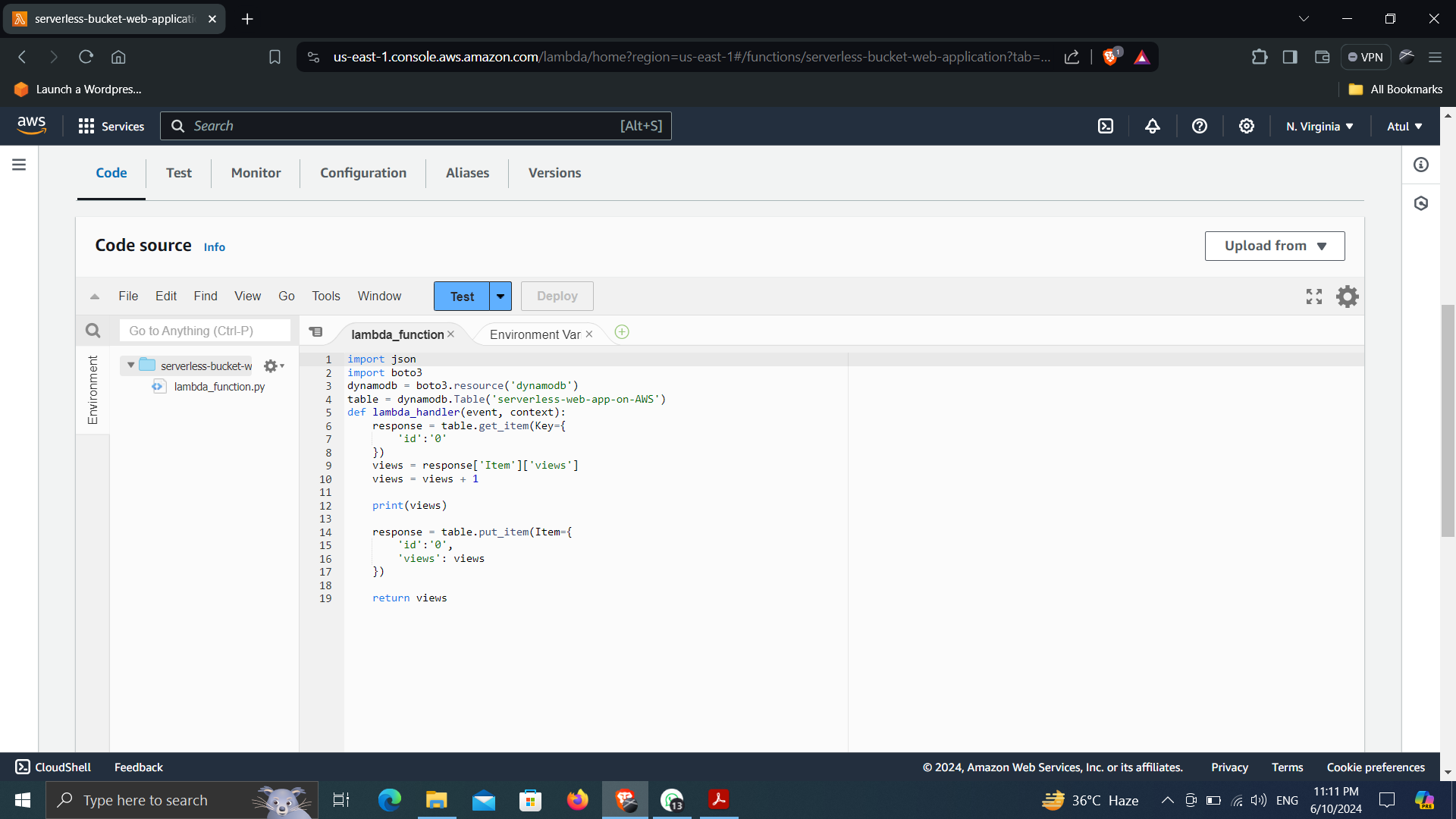
****

1. **In Lambda function, navigate to Configuration, then in permissions go to edit in execution role. Scroll down and go to existing role and choose IAM role and then save.**

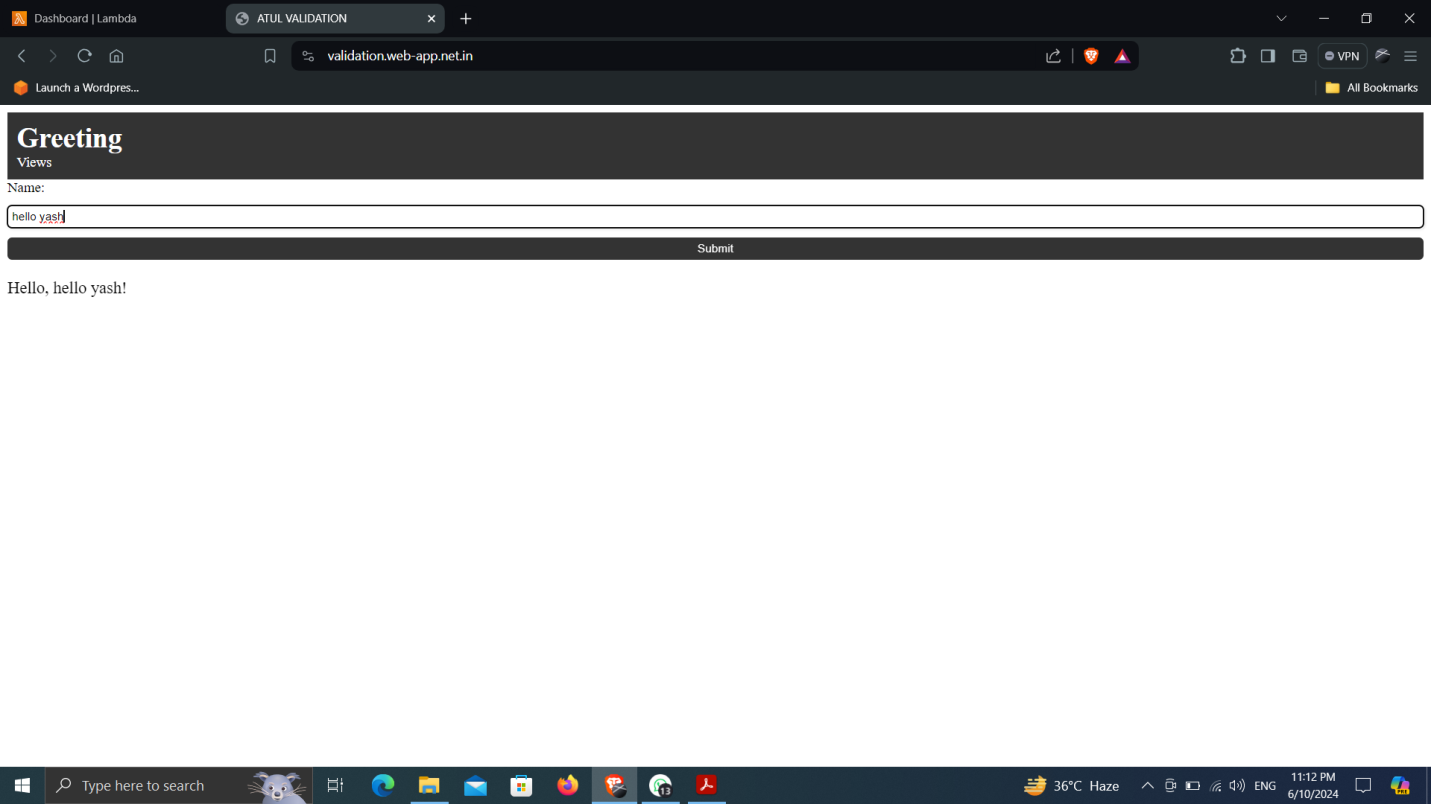
****

****

1. **Navigate to code section, then write the code for lambda function in aws editor or you can choose any editor and then you can paste your code to aws editor in lambda function. Then click on deploy and then Test. Check if there is any error or not. If error occur, then check your lambda function code.**

****

1. **Now check your website if it’s working successfully or not.**

****