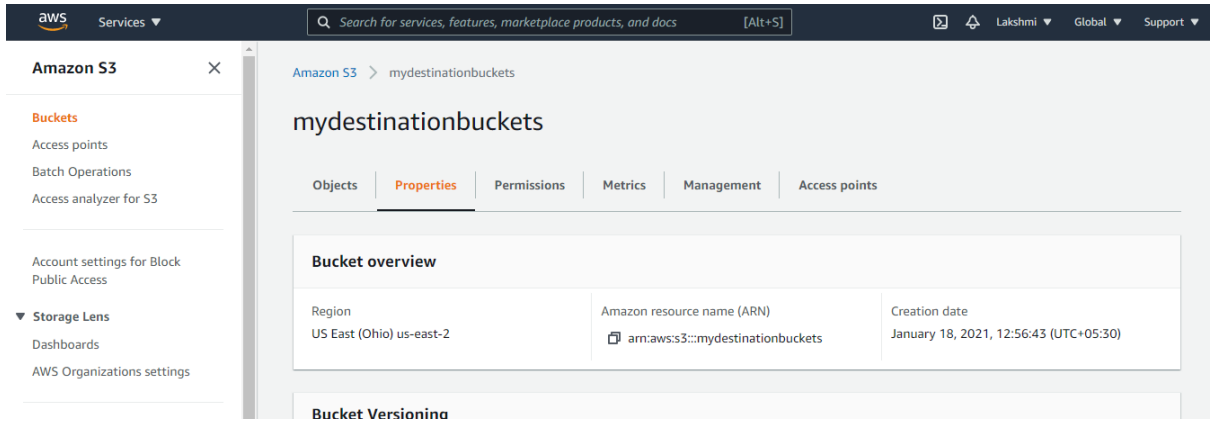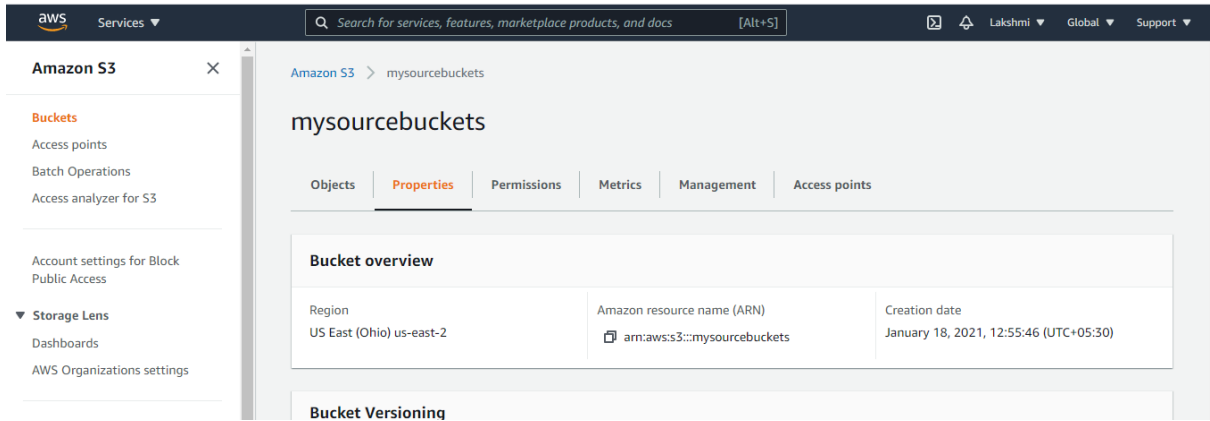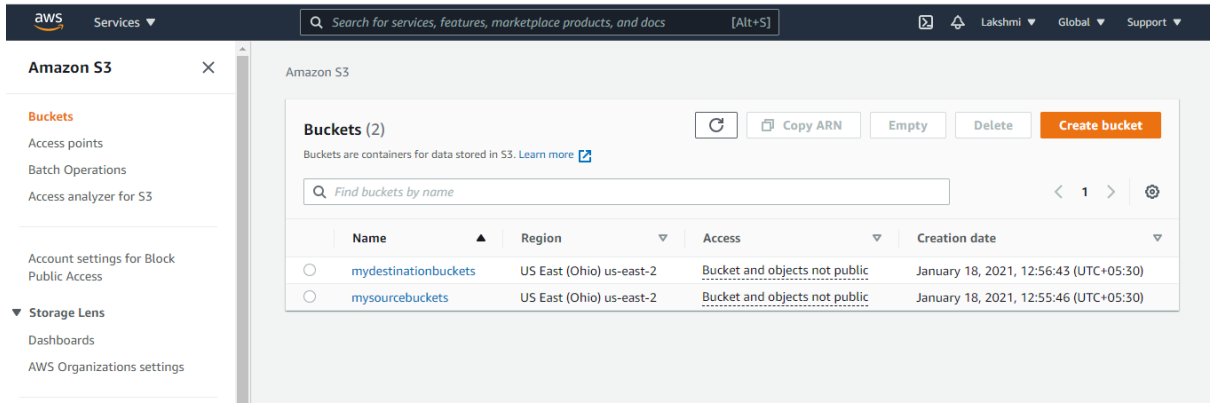**Working with Lambda**

Step1:Create two s3 buckets with the name

SS1:s3 console with two buckets

Step2:Creat a policy with limited Read-write permissions using a JSON script

ss2:json script in place


```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::mysourcebuckets/*"
            ]
        },
        {
```

ss3:policy console with your policy filtered

Step3:Create a role and attach the policy that was created in the previous step.

ss4:Role console showing details of the role

Step4:Create a Lambda function

ss5:lambda functions dashboard

ss6:js file edited

ss7:adding trigger-s3,bucket name,confirmation for having separate buckets
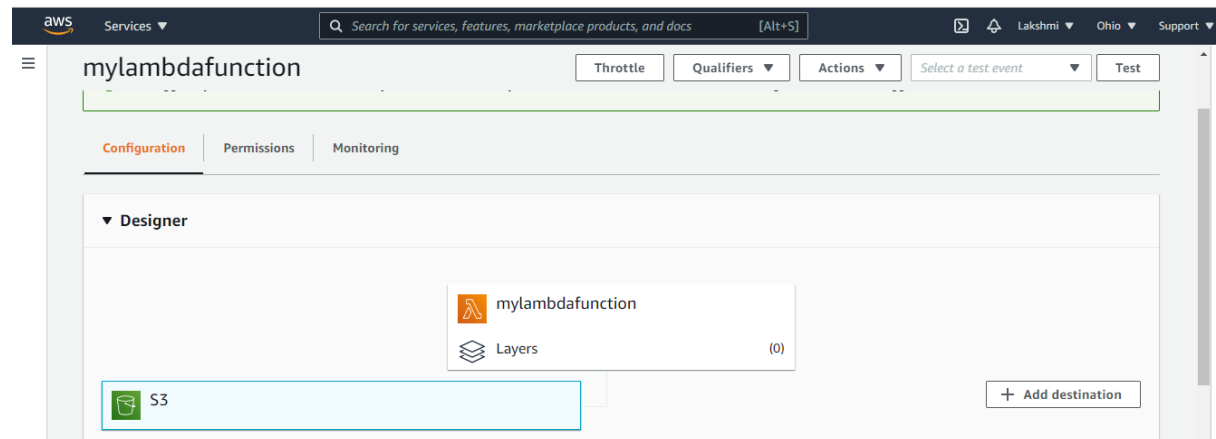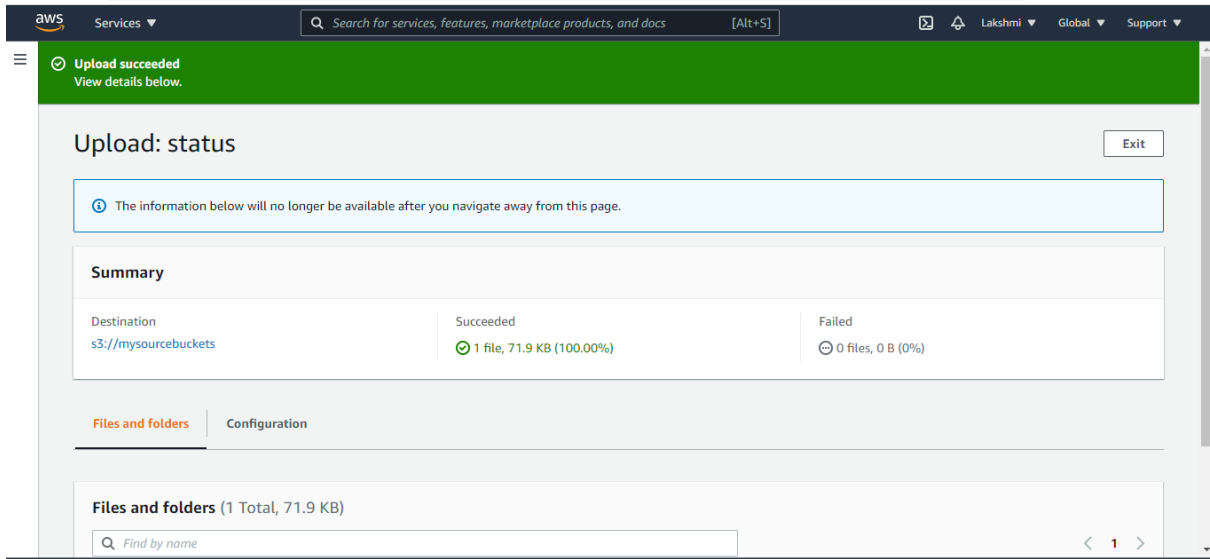
Step5:Adding triggers to the lambda function
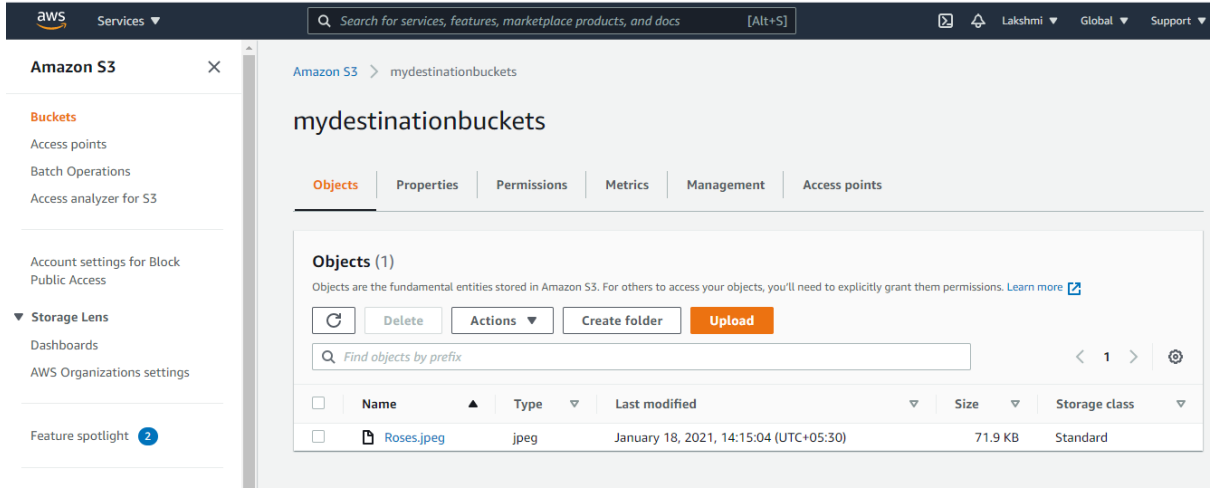
ss8:lambda configuration page with trigger added
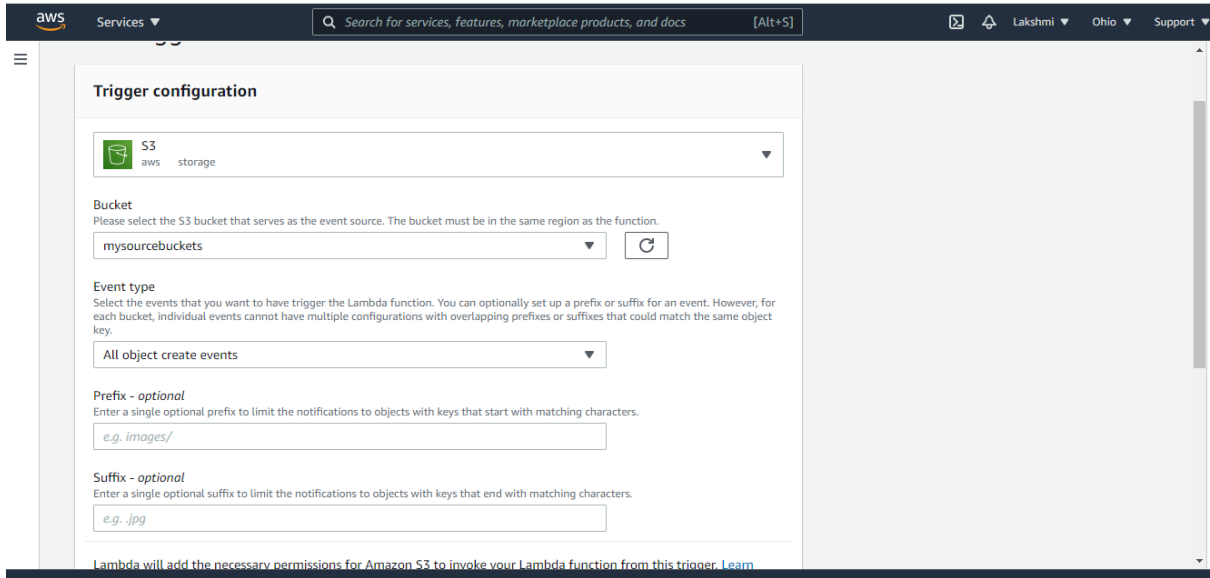
Step6:Test by uploading objects into the source bucket
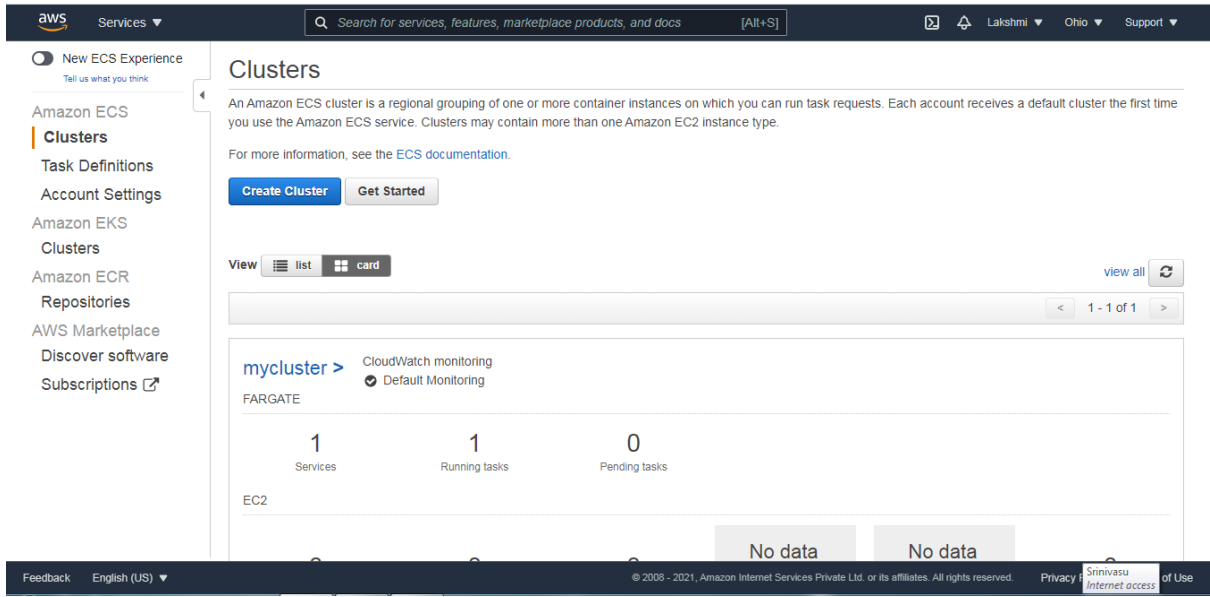
ss9:object uploaded in the source bucket

ss10:object replicated in the destination bucket.

Question 2: Working with Elastic container service using fargate

Step1:Getting started with amazon ECS using fargate

SS1:ECS console

Step2:Creating container and task definition

SS2:2nd panel with all options visible

Step3:Configuring the service

Step4:Configuring the cluster

Step5:viewing the service

ss5:Dashboard displaying the cluster created

ss6:cluster information

ss7:panel displaying ENI ID

ss8:Panel displaying the private, public, and the macid
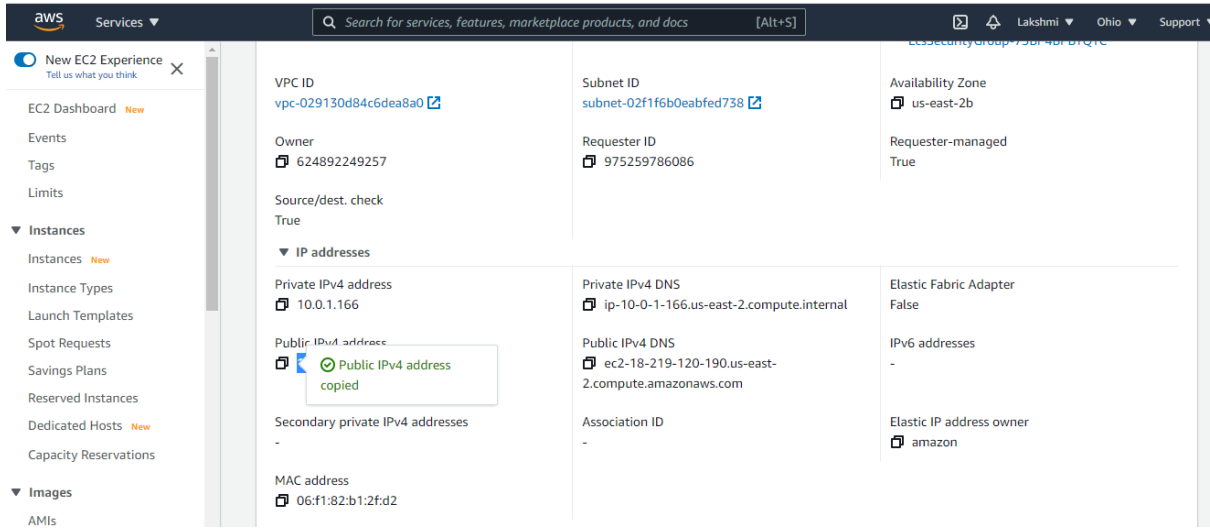
ss9:display application