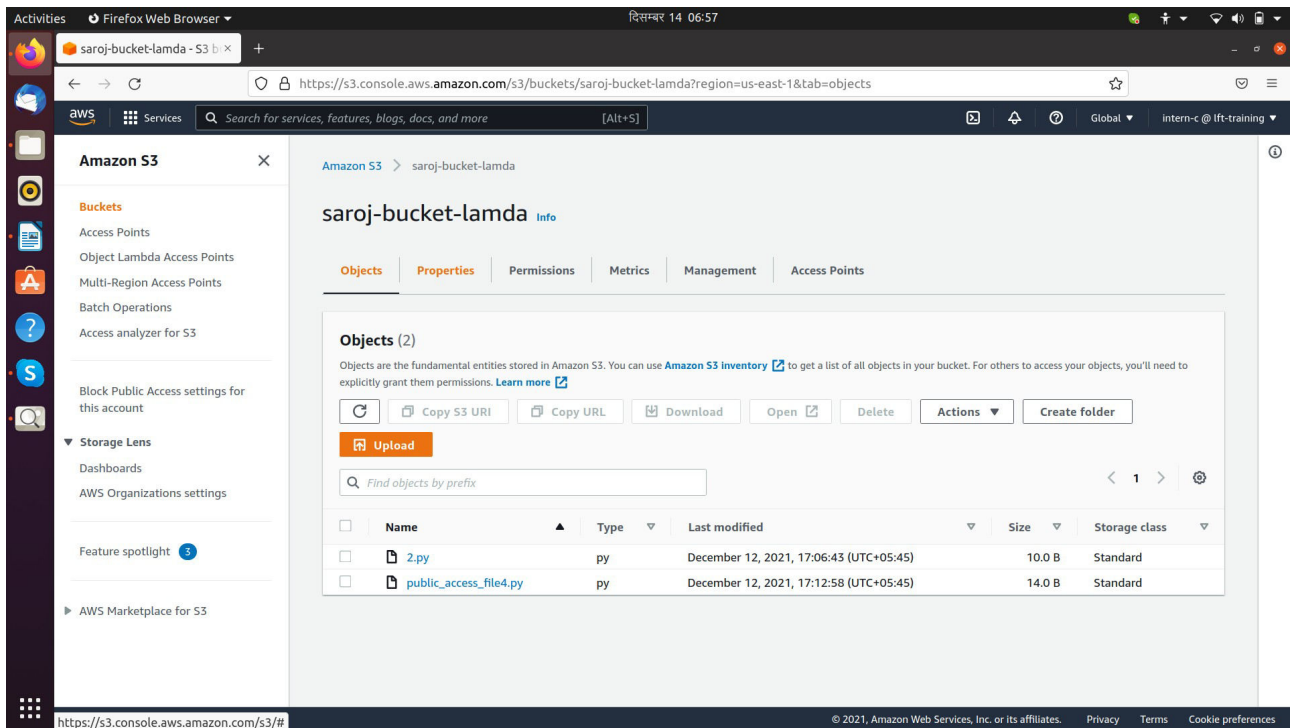


Task 2

Create a bucket (saroj-bucket-lamda)

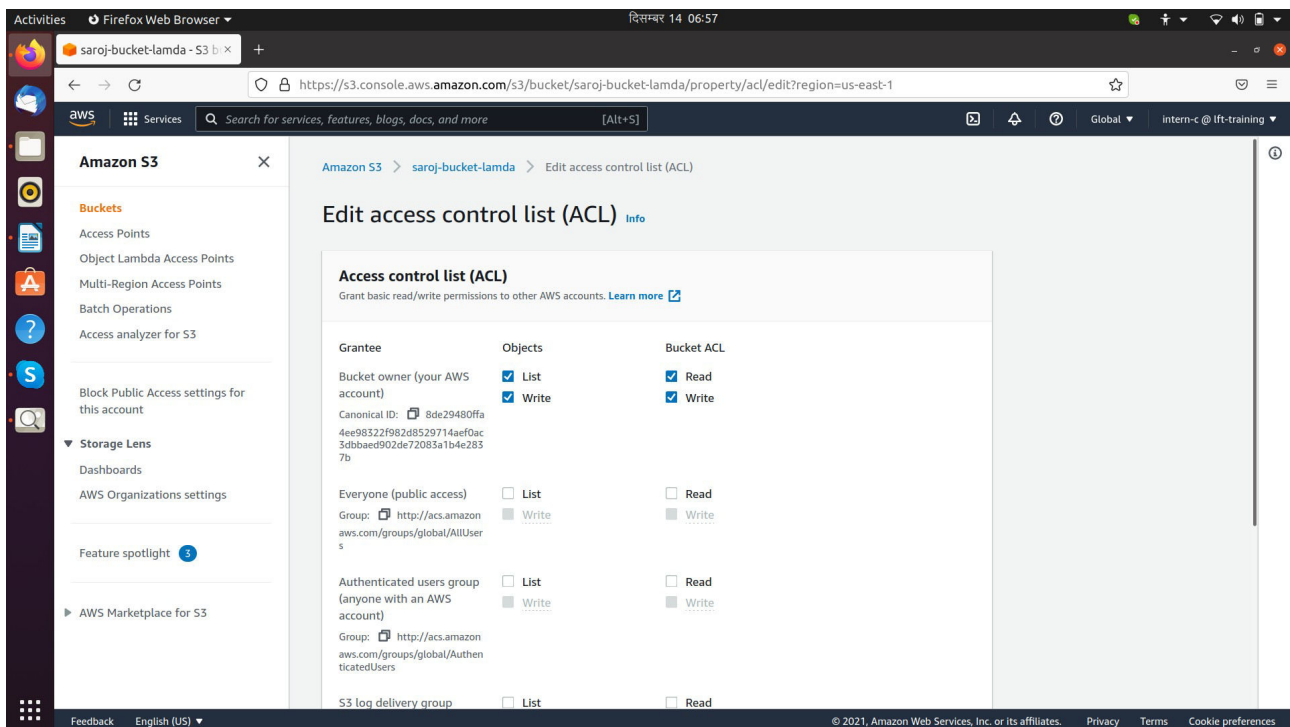


The screenshot shows the Amazon S3 console in a Firefox browser. The left sidebar contains navigation options like Buckets, Access Points, and Storage Lens. The main content area displays the 'saroj-bucket-lamda' bucket. Under the 'Objects (2)' section, there is a table listing two objects:

	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	2.py	py	December 12, 2021, 17:06:43 (UTC+05:45)	10.0 B	Standard
<input type="checkbox"/>	public_access_file4.py	py	December 12, 2021, 17:12:58 (UTC+05:45)	14.0 B	Standard

Below the table, there are buttons for 'Upload', 'Find objects by prefix', and a pagination control showing '1' of 1 objects.

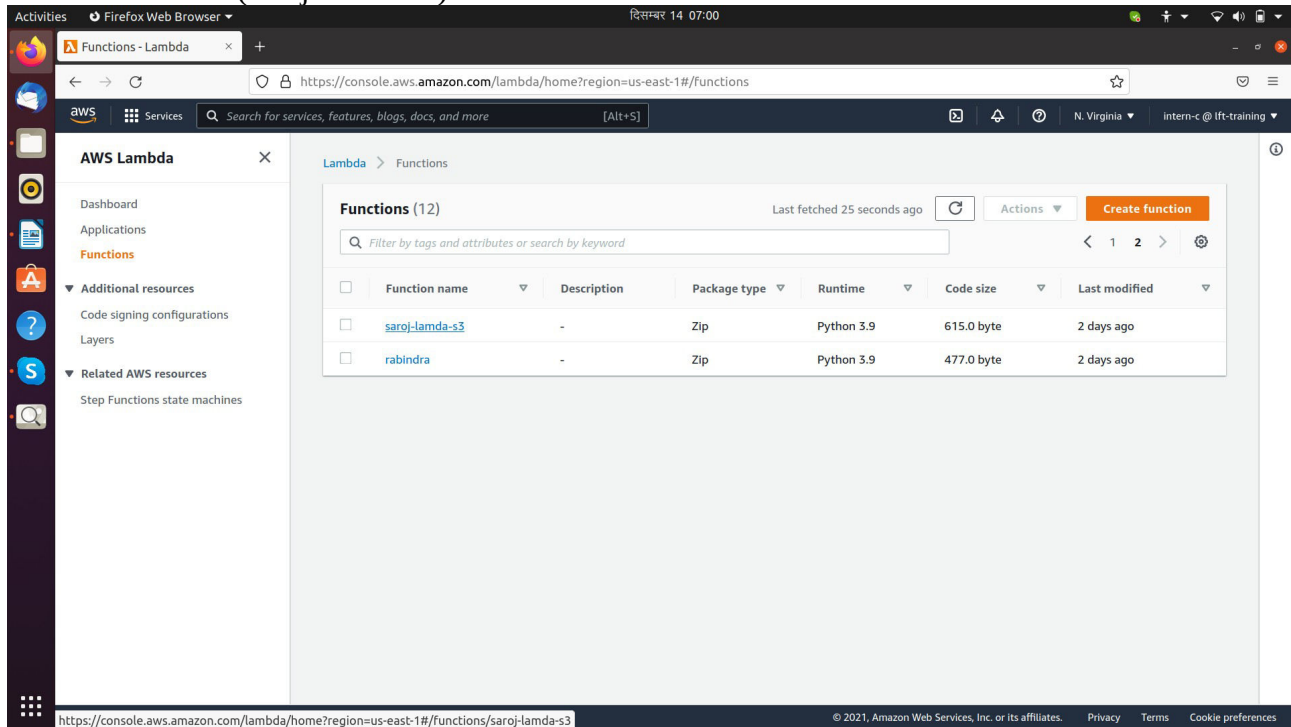
grant acl permission



The screenshot shows the 'Edit access control list (ACL)' page for the 'saroj-bucket-lamda' bucket. The page title is 'Edit access control list (ACL)'. Below the title, there is a section 'Access control list (ACL)' with a description: 'Grant basic read/write permissions to other AWS accounts. Learn more'. The table below lists the permissions for different grantees:

Grantee	Objects	Bucket ACL
Bucket owner (your AWS account) Canonical ID: 8de29480ffa4ee98322f982d8529714acf0ac3dbbaed902de72083a1b4e2837b	<input checked="" type="checkbox"/> List <input checked="" type="checkbox"/> Write	<input checked="" type="checkbox"/> Read <input checked="" type="checkbox"/> Write
Everyone (public access) Group: http://acs.amazonaws.com/groups/global/AllUsers	<input type="checkbox"/> List <input type="checkbox"/> Write	<input type="checkbox"/> Read <input type="checkbox"/> Write
Authenticated users group (anyone with an AWS account) Group: http://acs.amazonaws.com/groups/global/AuthenticatedUsers	<input type="checkbox"/> List <input type="checkbox"/> Write	<input type="checkbox"/> Read <input type="checkbox"/> Write
S3 log delivery group	<input type="checkbox"/> List	<input type="checkbox"/> Read

create a function (saroj-lambda-s3)



Add trigger s3 bucket
prefix public_access
event-type put

code

```
# import json
# import boto3
# import urllib

# s3 = boto3.client('s3')

# def lambda_handler(event, context):
#     #print("Received event: " + json.dumps(event, indent=2))
#     #s3 = boto3.client('s3')

#     # Get the object from the event and show its content type
#     bucket = event['Records'][0]['s3']['bucket']['name']
#     filename = event['Records'][0]['s3']['object']['key']

#     public_object=s3.Object(bucket, filename)

#     #get the ACL object for the uploaded object
#     acl=public_object.Acl()

#     #make the object publicly accessible
#     acl.put(ACL='public-read')

import json
```

```

import boto3

bucket = "saroj-bucket-lambda"

def lambda_handler(event, context):

    # extracting the bucket name and object name from the event
    bucket_name=event["Records"][0]["s3"]["bucket"]["name"]
    object_name=event["Records"][0]["s3"]["object"]["key"]

    print("bucket name is", bucket_name)
    print("object name is", object_name)

    #initializing an s3 resource
    s3=boto3.resource('s3')

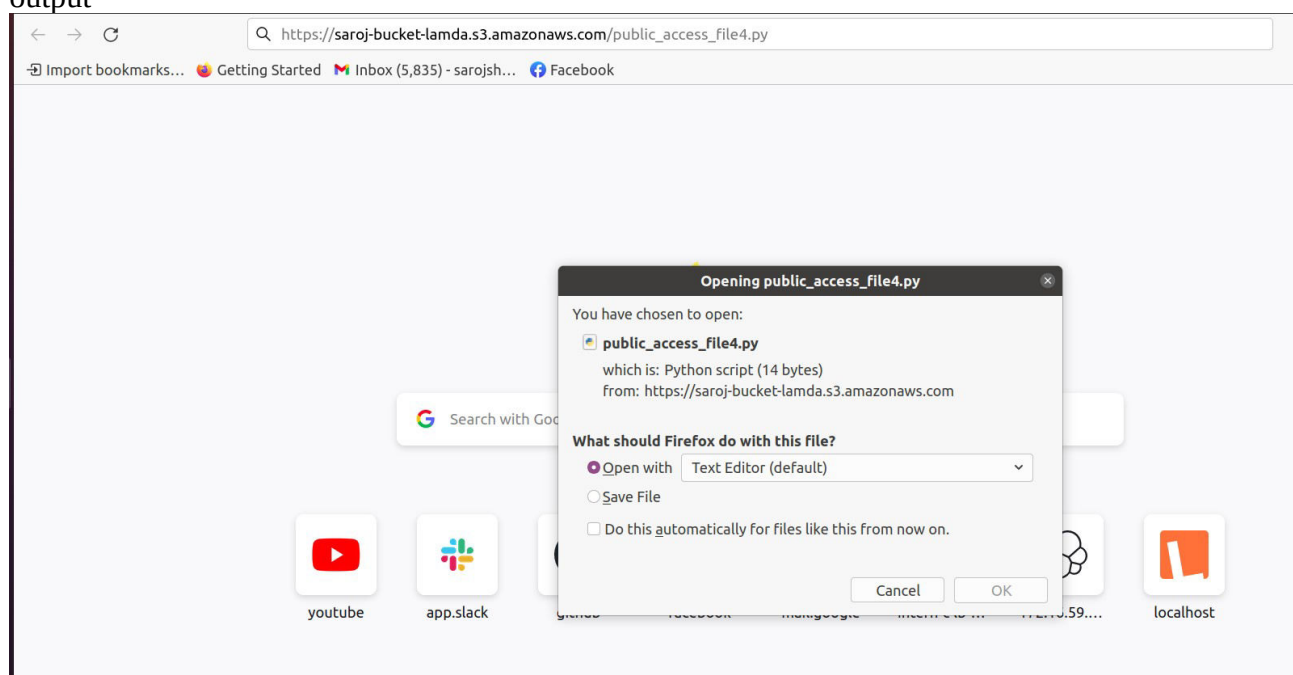
    #getting the uploaded public object
    public_object=s3.Object(bucket,object_name)

    #get the ACL object for the uploaded object
    acl=public_object.Acl()

    #make the object publicly accessible
    acl.put(ACL='public-read')

```

output





<https://saroj-bucket-lambda.s3.amazonaws.com/2.py>

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
--<Error>
  <Code>AccessDenied</Code>
  <Message>Access Denied</Message>
  <RequestId>97EJ8CYV1EK46HTP</RequestId>
--<HostId>
  eB/+IrPaa9NbvS7GUTWNc8o0h67iw30thAnuzB46gl8m8qmLXPfWK88Ky9VpEovZRqzqXef6sp0=
</HostId>
</Error>
```