

AWS TASK LIST

SR/NO	TASK
1	create user assign one policy (ec2 full permission) but any specific region(mumbai)
2	create bucket and apply mfa to the bucket
3	create policy that given root user access to another user for specific time after specific time access get denied if he try to access get access denied error
4	upload object in bucket this object is private but other user can access it only for specific time
5	create policy and give user to access or use only one specific bucket read and write list permission
6	ec2 instance name:- scripted-web-hosting
7	Ec2 Unblock visibility Public Access for AMIs
8	Share an EC2 AMI with a Friend's AWS Account.
9	ec2 :- EC2 recover loss key-pair .

Task No:1

Name: sagar

* Ec2-to grant region specific permation allow only ap-south-1a

//create policy

The screenshot shows the AWS Policy Generator interface. It is titled "AWS Policy Generator" and provides instructions on how to create policies. The user has selected "IAM Policy" as the policy type. In the "Step 2: Add Statement(s)" section, the user has configured a statement with the following details:

- Effect:** Allow
- AWS Service:** Amazon EC2
- Actions:** -- Select Actions --
- Amazon Resource Name (ARN):** (Empty field)

The user has added a condition: "StringNotEqualsIfExists: aws:RequestedRegion: 'ap-south-1'". The "Add Statement" button is visible.

Below the configuration, a table shows the generated statements:

Effect	Action	Resource	Conditions
Deny	*	*	StringNotEqualsIfExists aws:RequestedRegion: "ap-south-1"
Allow	*	*	None

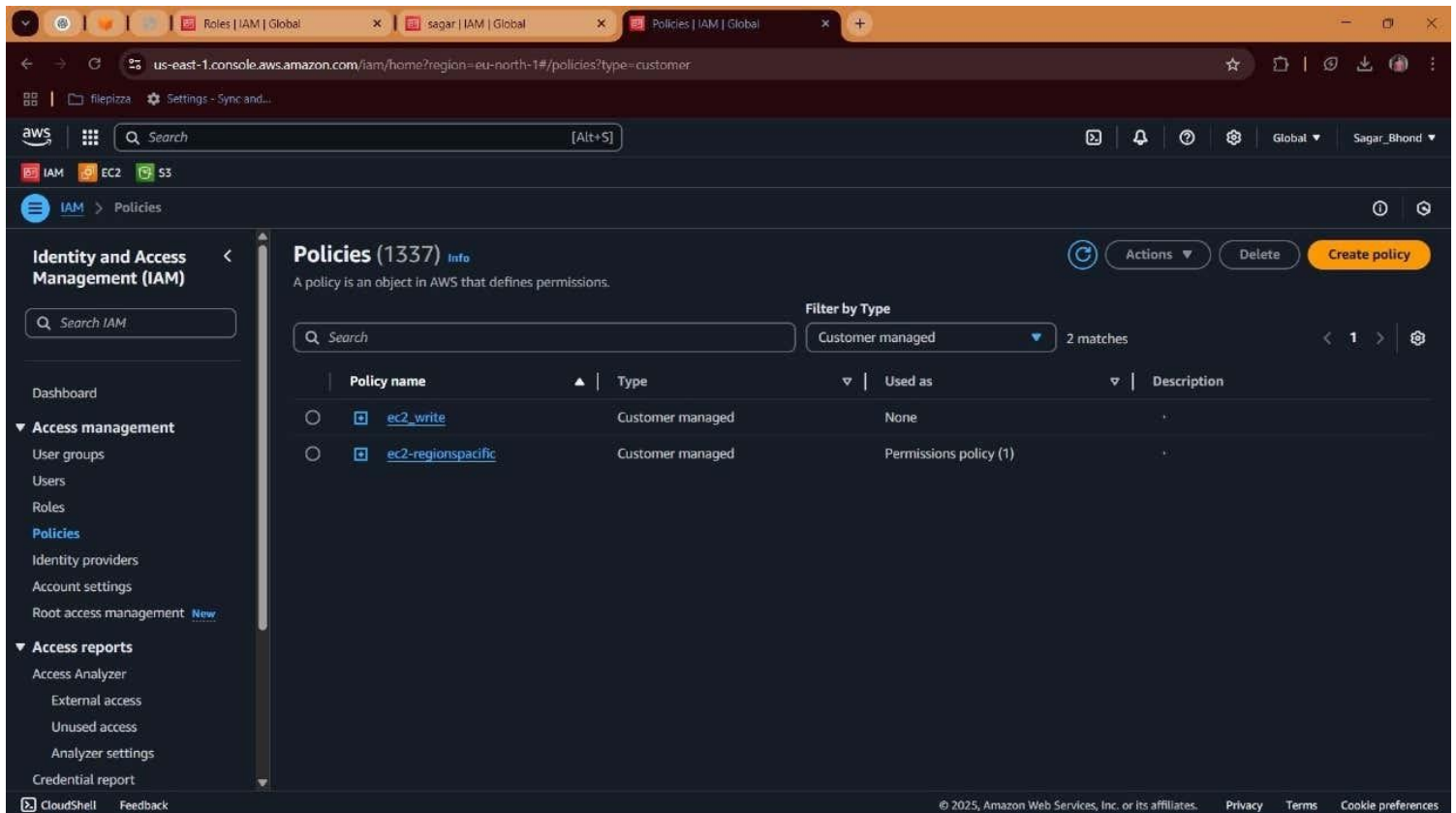
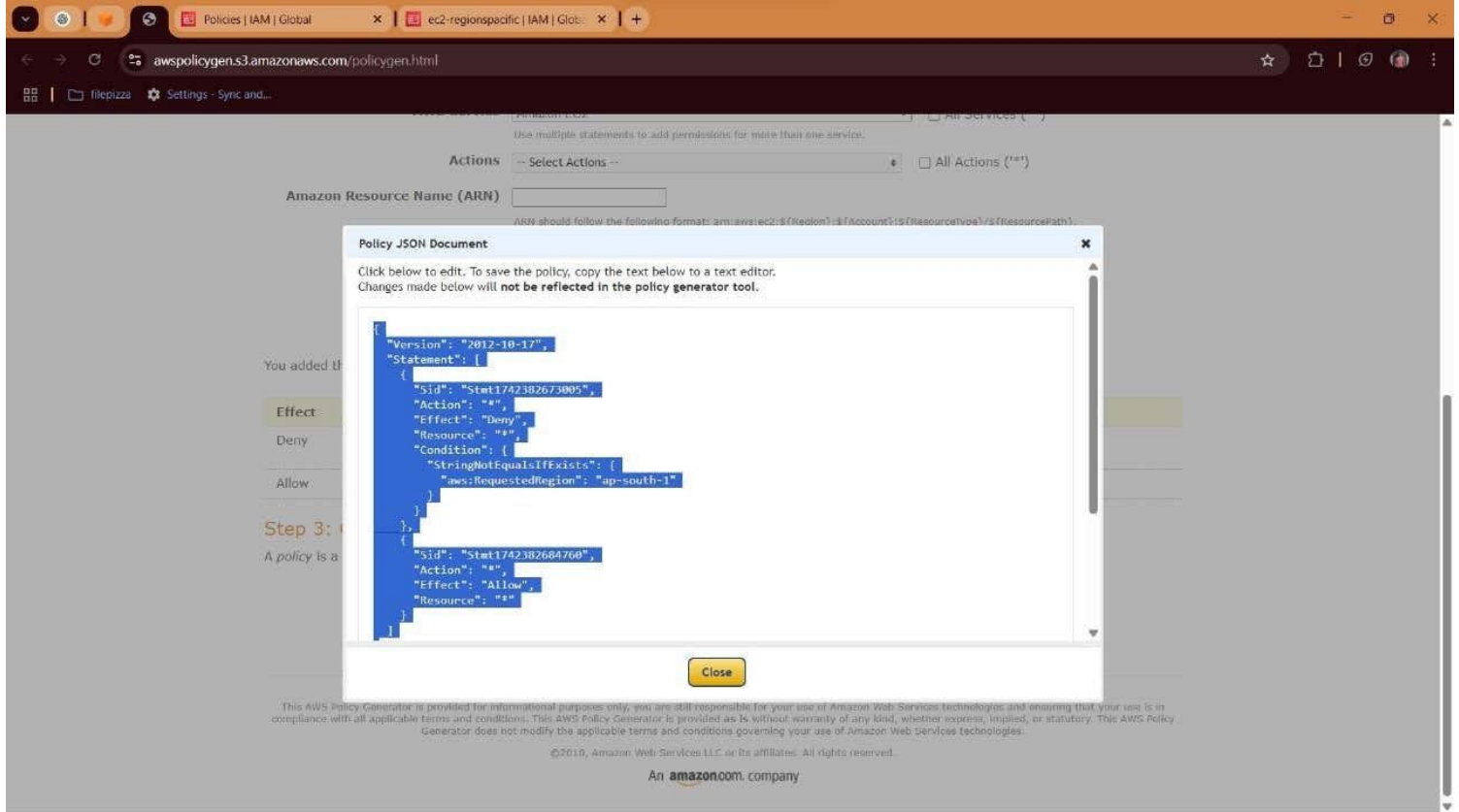
The user has added the following statements. Click the button below to Generate a policy.

Step 3: Generate Policy

A policy is a document (written in the Access Policy Language) that acts as a container for one or more statements.

Generate Policy **Start Over**

This AWS Policy Generator is provided for informational purposes only, you are still responsible for your use of Amazon Web Services technologies and ensuring that your use is in compliance with all applicable terms and conditions. This AWS Policy Generator is provided as is without warranty of any kind, whether express, implied, or statutory. This AWS Policy Generator does not modify the applicable terms and conditions governing your use of Amazon Web Services technologies.



Launch an instance | EC2 | ap-south-1

Instances | EC2 | ap-south-1

ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#InstancesinstanceId=i-0654d686ecdcf74f2

filepizza Settings - Sync and...

aws Search [Alt+S]

Asia Pacific (Mumbai) sagar @ 1205-6962-4603

EC2 > Instances

EC2

Dashboard

EC2 Global View

Events

Instances

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity Reservations

Images

AMIs

AMI Catalog

Elastic Block Store

Volumes

Snapshots

Lifecycle Manager

Instances (1) Info

Last updated less than a minute ago

Connect

Instance state

Actions

Launch instances

Find Instance by attribute or tag (case-sensitive)

Instance ID = i-0654d686ecdcf74f2

Clear filters

1

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
<input type="checkbox"/>	new	i-0654d686ecdcf74f2	Running	t2.micro	Initializing	View alarms +	ap-south-1a	ec2-13-232

Select an instance

CloudShell Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Launch an instance | EC2 | us-east-1

Instances | EC2 | us-east-1

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#InstancesinstanceId=i-0654d686ecdcf74f2

filepizza Settings - Sync and...

aws Search [Alt+S]

United States (N. Virginia) sagar @ 1205-6962-4603

EC2 > Instances

EC2

Dashboard

EC2 Global View

Events

Instances

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity Reservations

Images

AMIs

AMI Catalog

Elastic Block Store

Volumes

Snapshots

Lifecycle Manager

Instances Info

Last updated less than a minute ago

Connect

Instance state

Actions

Launch instances

Find Instance by attribute or tag (case-sensitive)

Instance ID = i-0654d686ecdcf74f2

Clear filters

1

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
<div>You are not authorized to perform this operation. User: arn:aws:iam::120569624603:user/sagar is not authorized to perform: ec2:DescribeInstances with an explicit deny in an identity-based policy</div>								

Select an instance

CloudShell Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Task No:2

Name: sagar

*AWS-MFA on S3 - Notepad

Step:1 \$ aws configure

Step:2 \$ aws s3api list-buckets

//check bucket list

```
sagar_c7otr@SagarBhond MINGW64 ~
$ aws s3api list-buckets
{
  "Buckets": [
    {
      "Name": "sagar-007",
      "CreationDate": "2025-03-23T18:46:36+00:00"
    }
  ],
  "Owner": {
    "ID": "8d1069d9c60f0fd3a07a44000b77dfdc4c68252f1a82a58f66b1eae3da53b79b"
  },
  "Prefix": null
}
```

Step:3 \$ aws s3api get-bucket-versioning --bucket sagar-007

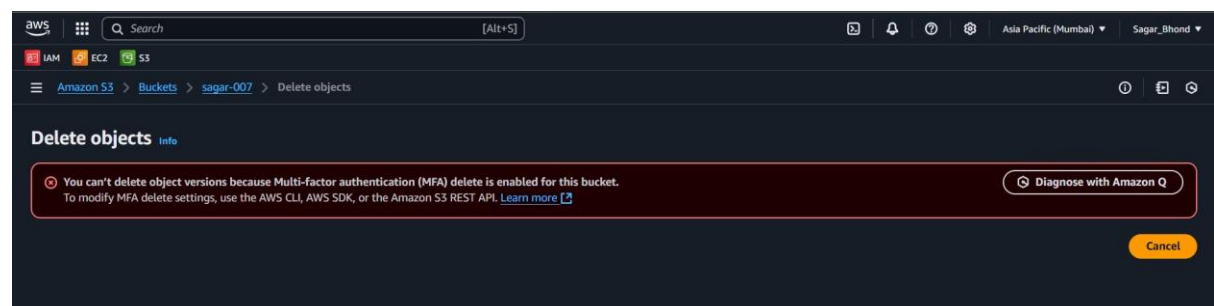
//check versioning is on or off

```
sagar_c7otr@SagarBhond MINGW64 ~
$ aws s3api get-bucket-versioning --bucket sagar-007
{
  "Status": "Enabled",
  "MFADelete": "Disabled"
}
```

Step:4 \$ aws s3api put-bucket-versioning --bucket sagar-007 --versioning-configuration Status=Enabled,MFADelete=Enabled --mfa "arn:aws:iam::120569624603:mfa/rootuser 270127"

//enable mfa providing root arn and root mfa code

```
C:\Users\sagar_c7otr>aws s3api put-bucket-versioning --bucket sagar-007 --versioning-configuration Status=Enabled,MFADelete=Enabled --mfa "arn:aws:iam::120569624603:mfa/rootuser 270127"
```



//if you want to delete file

Step:1 \$ aws s3api delete-object -- bucket sagar-007 -- key .jpg

(File Was deleted because it protect only Versionong File)

Step:2 \$ aws s3api delete-object -- bucket sagar-007 -- key .jpg -- version-id Eqv102ILFiiBJPL2kCWVUBOXPSw>

(now Try to delet VersioniD File It required MFA)

Step:3 \$ aws s3api put-bucket-versioning -- bucket sagar-007 -- versioning-configuration Status=Enabled, MFADelete=Disab]

Note: - It is very in-secure methos because your "access key" & "secrate key" store on locally.

Anyone can access your laptop can access your aws keys.

Versioning must be enable on Bucket.

Task No:3

Name: sagar

* IAMAutoStopPolicy- after 10 minute.

Step:1 create user sai.

Step:2 create policy to logout sai user after 10 minute.

Step 1: Select Policy type

A Policy is a container for permissions. The different types of policies you can create are an IAM Policy, an S3 Bucket Policy, an SNS Topic Policy, a VPC Endpoint Policy, and an SQS Queue Policy.

Select Type of Policy

Step 2: Add Statement(s)

A statement is the formal description of a single permission. See a description of elements that you can use in statements.

Effect ☒ Allow ☐ Deny

AWS Service ☐ All Services ("*")

Use multiple statements to add permissions for more than one service.

Actions ☐ All Actions ("*")

Amazon Resource Name (ARN)

ARN should follow the following format: arn:aws:ec2:\${Region}:\${Account}:\${ResourceType}/\${ResourcePath}.
Use a comma to separate multiple values.

Add Conditions (Optional)

You added the following statements. Click the button below to Generate a policy.

Effect	Action	Resource	Conditions
Allow	*	*	<ul style="list-style-type: none">DateGreaterThan<ul style="list-style-type: none">aws:CurrentTime: "2025-03-26T11:42:07+05:30"DateLessThanEquals<ul style="list-style-type: none">aws:CurrentTime: "2025-03-26T12:02:08+05:30"

Step 3: Generate Policy

A policy is a document (written in the Access Policy Language) that acts as a container for one or more statements.

Policy JSON Document

Click below to edit. To save the policy, copy the text below to a text editor.
Changes made below will **not be reflected in the policy generator tool**.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1742969726570",
      "Action": "*",
      "Effect": "Allow",
      "Resource": "*",
      "Condition": {
        "DateGreaterThan": {
          "aws:CurrentTime": "2025-03-26T11:42:07+05:30"
        },
        "DateLessThanEquals": {
          "aws:CurrentTime": "2025-03-26T12:02:08+05:30"
        }
      }
    }
  ]
}
```


Step:3 attach that policy to sai user

sai info Delete

Summary

ARN
arn:aws:iam::120569624603:user/sai

Console access
Enabled without MFA

Access key 1
[Create access key](#)

Created
March 26, 2025, 11:27 (UTC+05:30)

Last console sign-in
Today

Permissions Groups Tags Security credentials Last Accessed

Permissions policies (1) Remove Add permissions

Permissions are defined by policies attached to the user directly or through groups.

Filter by Type
All types

Search

Policy name Policy name | Type | Attached via

EC2AutoStopPolicy Customer managed Directly

Permissions boundary (not set)

Step:4 login sai user with 1) username = sai 2) password sai@1234

//after 10 minute it automatically logout

My security credentials info

Manage credentials for your currently authenticated IAM user. To learn more about the types of AWS credentials and how they're used, see [AWS Security Credentials](#).

Account details

User name
sai

AWS account ID
120569624603

AWS IAM credentials

Console sign-in

Console sign-in link
<https://120569624603.signin.aws.amazon.com/console>

Console password

Access denied
You don't have permission to iam:GetLoginProfile. To request access, copy the following text and send it to your AWS administrator. [Learn more about troubleshooting access denied errors.](#)

User: arn:aws:iam::120569624603:user/sai
Action: iam:GetLoginProfile
Context: no identity-based policy allows the action

You have been signed out
You've been signed out of the AWS console. AWS sessions in all tabs have been signed out.
Sign in again

Users (0) info Delete Create user

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

Search

User name | Path | Group | Last activity | MFA | Password age | Console last sign-in | Access key

Access denied
You don't have permission to iam:ListUsers. To request access, copy the following text and send it to your AWS administrator. [Learn more about troubleshooting access denied errors.](#)

User: arn:aws:iam::120569624603:user/sai
Action: iam:ListUsers
On resource(s): arn:aws:iam::120569624603:user/
Context: no identity-based policy allows the action

[Diagnose with Amazon Q](#)

Task No:4

Name: sagar

* Grant Permission Via Json s3AutoStopPolicy- after 3 hr.

Step:1 create user [s3-logout3hr](#).

Step:2 create policy to Grant Permission through json.

VPC Endpoint Policy, and an SQS Queue Policy.

Select Type of Policy

Step 2: Add Statement(s)

A statement is the formal description of a single permission. See a description of elements that you can use in statements.

Effect ☒ Allow ☐ Deny

AWS Service ☐ All Services ("*")

Use multiple statements to add permissions for more than one service.

Actions ☐ All Actions ("*")

Amazon Resource Name (ARN)

ARN should follow the following format: arn:aws:s3:::{BucketName}/{Keyname}.
Use a comma to separate multiple values.

Add Conditions (Optional)

You added the following statements. Click the button below to Generate a policy.

Effect	Action	Resource	Conditions
Allow	s3:GetObject	arn:aws:s3:::s3-logoutafter-3hor/sagar_Resume.pdf	<ul style="list-style-type: none">DateGreaterThanEquals<ul style="list-style-type: none">aws:CurrentTime: "2025-03-26T13:50:00+05:30"DateLessThanEquals<ul style="list-style-type: none">aws:CurrentTime: "2025-03-26T16:50:00+05:30"

Step 3: Generate Policy

A policy is a document (written in the Access Policy Language) that acts as a container for one or more statements.

Actions ☐ All Actions ("*")

Amazon Resource Name (ARN)

ARN should follow the following format: arn:aws:s3:::{BucketName}/{Keyname}.
Use a comma to separate multiple values.

You added the

Effect

Allow

Step 3: Generate Policy

A policy is a

Policy JSON Document

Click below to edit. To save the policy, copy the text below to a text editor.
Changes made below will not be reflected in the policy generator tool.

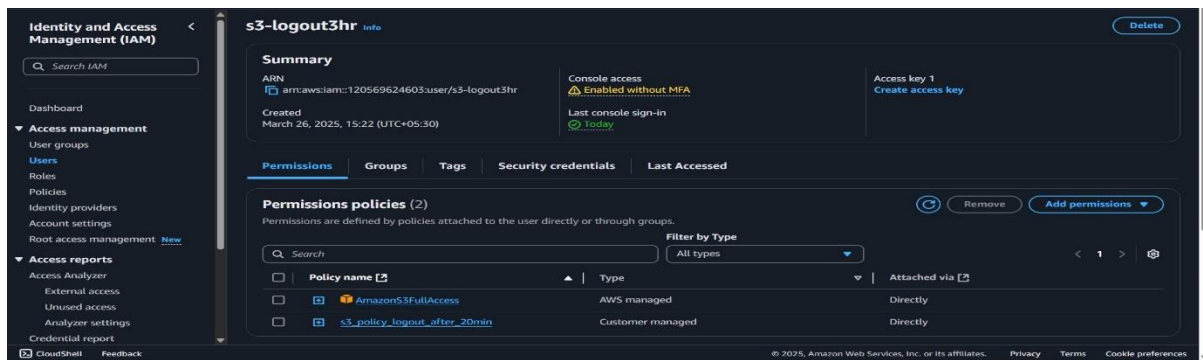
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1743064027115",
      "Action": [
        "s3:GetObject"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::s3-logoutafter-3hor/sagar_Resume.pdf",
      "Condition": {
        "DateGreaterThanEquals": {
          "aws:CurrentTime": "2025-03-26T13:50:00+05:30"
        },
        "DateLessThanEquals": {
          "aws:CurrentTime": "2025-03-26T16:50:00+05:30"
        }
      }
    }
  ]
}
```

This AWS Policy Generator is provided for informational purposes only; you are still responsible for your use of Amazon Web Services technologies and ensuring that your use is in compliance with all applicable terms and conditions. This AWS Policy Generator is provided as is without warranty of any kind, whether express, implied, or statutory. This AWS Policy Generator does not modify the applicable terms and conditions governing your use of Amazon Web Services technologies.

©2010, Amazon Web Services LLC or its affiliates. All rights reserved.

An amazon.com company

Step:3 attach that policy to **s3-logoutafter-3hor** user

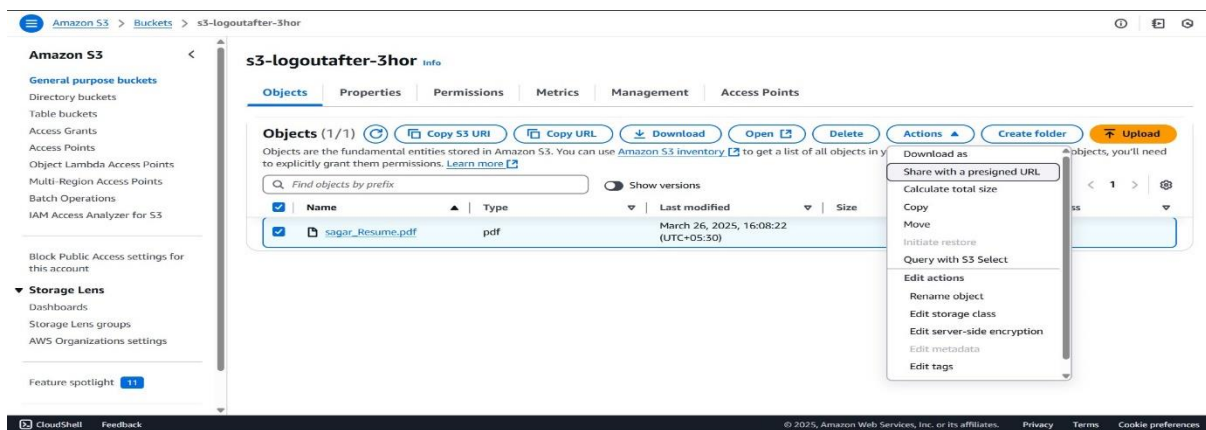


Step:4 login s3-logoutafter-3hor user with 1) username = s3-logoutafter-3hor 2) password **s3-logoutafter-3hor**

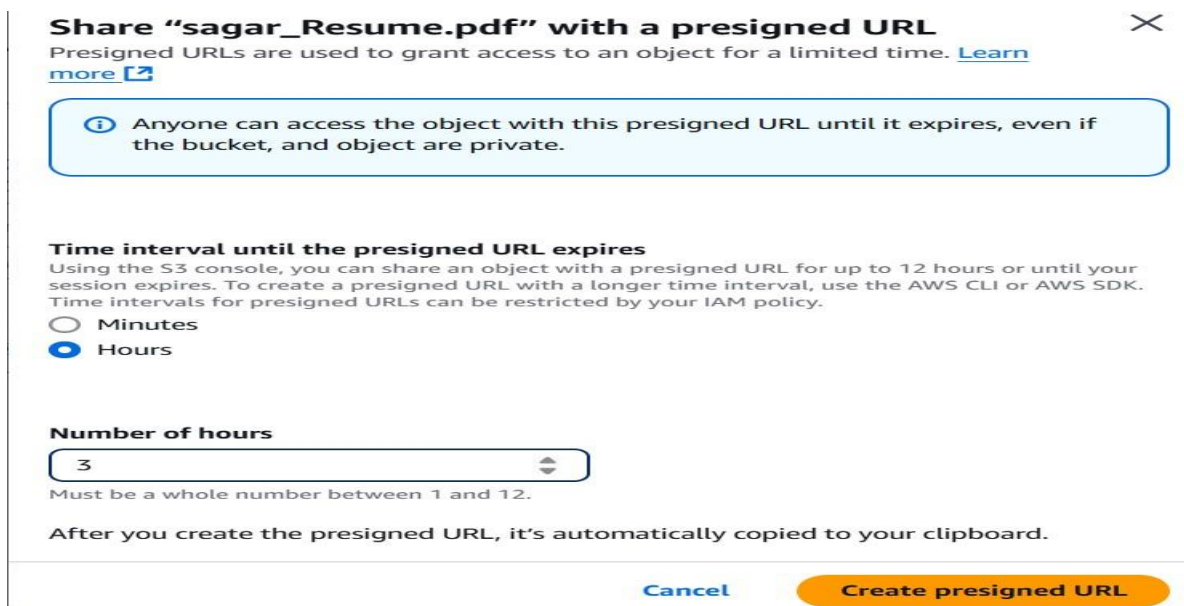
//create bucket name as s3-logoutafter-3hor

//uplode your resume on their

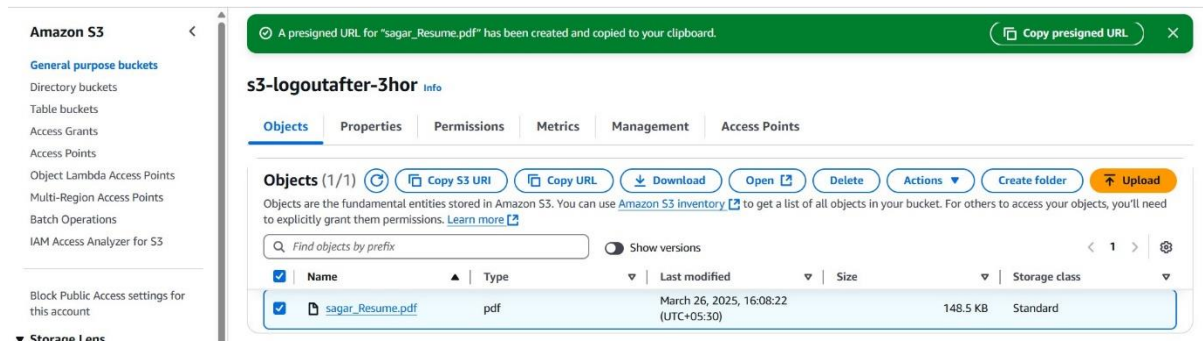
//select the resume click -> action -> share with a presigned url



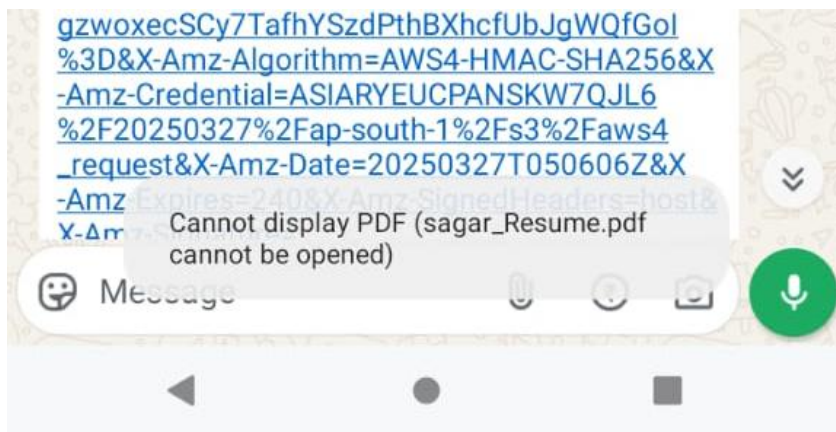
//set time of 3 hour



// after that it generate presigned url send this url to other



//after 3 hour while clicking this link it gives error like cannot display pdf (sagar_resume.pdf cannot be opened)



Task No:5

Name: sagar

* S3-to grant specific user permission-read,write,list.

Step:1 create user wiper.

Step:2 create policy.

Select Type of Policy IAM Policy

Step 2: Add Statement(s)

A statement is the formal description of a single permission. See a description of elements that you can use in statements.

Effect ☒ Allow ☐ Deny

AWS Service Amazon S3 ☐ All Services ("*")

Use multiple statements to add permissions for more than one service.

Actions -- Select Actions -- ☐ All Actions ("*")

Amazon Resource Name (ARN)

ARN should follow the following format: arn:aws:s3::s(BucketName)/s(KeyName).
Use a comma to separate multiple values.

Add Conditions (Optional)

Add Statement

You added the following statements. Click the button below to Generate a policy.

Effect	Action	Resource	Conditions
Allow	<ul style="list-style-type: none">s3:ListBucket	arn:aws:s3::sagar-009	None
Allow	<ul style="list-style-type: none">s3:GetObjects3:PutObject	arn:aws:s3::sagar-009/*	None

Step 3: Generate Policy

A policy is a document (written in the Access Policy Language) that acts as a container for one or more statements.

Generate Policy **Start Over**

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1743094718987",
      "Action": [
        "s3:ListBucket"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3::sagar-009"
    },
    {
      "Sid": "Stmt1743094779514",
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3::sagar-009/*"
    }
  ]
}
```

Close

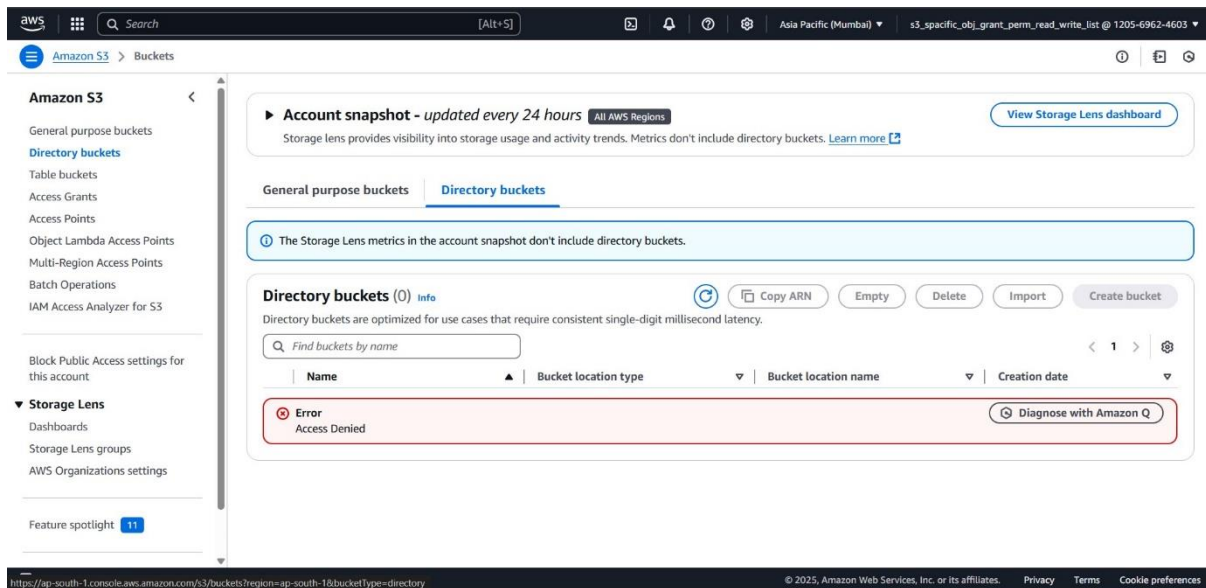
Step:3 give access to only s3_specific_obj_grant_perm_read_write_list.

The screenshot shows the AWS IAM console for a user named `s3_specific_obj_grant_perm_read_write_list`. The left sidebar shows the navigation menu with 'Access management' expanded. The main content area shows the 'Permissions' tab. The 'Summary' section displays the user's ARN, console access status (Enabled without MFA), and last console sign-in. The 'Permissions policies' section shows two policies: `AmazonS3FullAccess` (AWS managed) and `s3_specific_obj_grant_perm_read_write_list` (Customer managed), both attached directly. The 'Permissions boundary' section shows it is not set.

//previously when I give only s3fullacces it look like this type

The screenshot shows the AWS S3 console for the `s3_specific_obj_grant_perm_read_write_list` user. The left sidebar shows the navigation menu with 'Storage Lens' expanded. The main content area shows the 'General purpose buckets' tab. The 'Account snapshot' section shows the account is updated every 24 hours. The 'General purpose buckets' section shows a list of three buckets: `s3-logoutafter-3hor`, `sagar-007`, and `sagar-009`, all in the Asia Pacific (Mumbai) region. The 'Create bucket' button is visible.

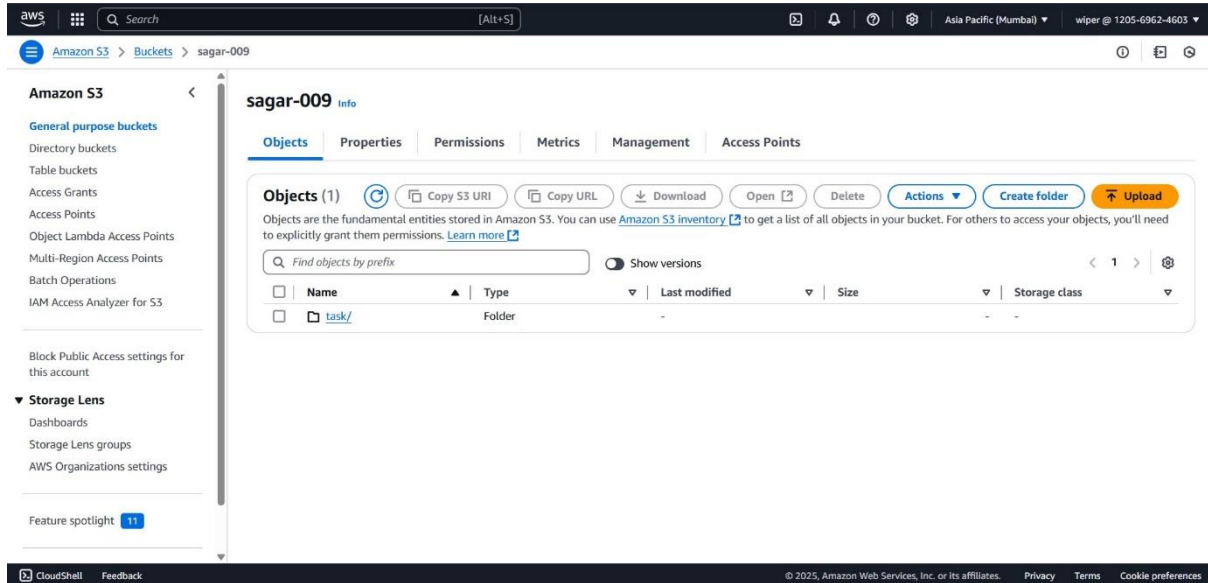
//after I add only s3_specific_obj_grant_perm_read_write_list it look like.



//edit the url bucket name in which bucket did you apply permission.



//after that it give the sagar-009 bucket access.



aws

Search

[Alt+S]

Asia Pacific (Mumbai)

wiper @ 1205-6962-4603

Amazon S3

Buckets

sagar-009

task/

Amazon S3

General purpose buckets

Directory buckets

Table buckets

Access Grants

Access Points

Object Lambda Access Points

Multi-Region Access Points

Batch Operations

IAM Access Analyzer for S3

Block Public Access settings for this account

Storage Lens

Dashboards

Storage Lens groups

AWS Organizations settings

Feature spotlight 11

task/

Copy S3 URI

Objects (4)

Copy S3 URI

Copy URL

Download

Open

Delete

Actions

Create folder

Upload

Find objects by prefix

Show versions

1

Table with 7 columns: Name, Type, Last modified, Size, Storage class. Rows include Task No 1.pdf, Task No 4.pdf, Task_No_3.pdf, Task_No_2.pdf.

CloudShell

Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Task No:6

Name: sagar

ec2 instance name:- scripted-web-hosting

//after that Allow HTTP traffic from the internet

// go to Advanced details option

// give key pair name san

// write User data - optional script

The screenshot shows the AWS Management Console 'Launch an instance' page. The breadcrumb navigation is 'EC2 > Instances > Launch an instance'. The 'Allow tags in metadata' dropdown is set to 'Select'. The 'User data - optional' section has a 'Choose file' button and a text area containing the following script:

```
#!/bin/bash
sudo yum install httpd -y
sudo systemctl start httpd
sudo systemctl enable httpd
sudo curl -O https://www.free-css.com/assets/files/free-css-templates/download/page296/carvillia.zip
sudo unzip carvillia.zip
sudo mv carvillia-v1.0/* /var/www/html
```

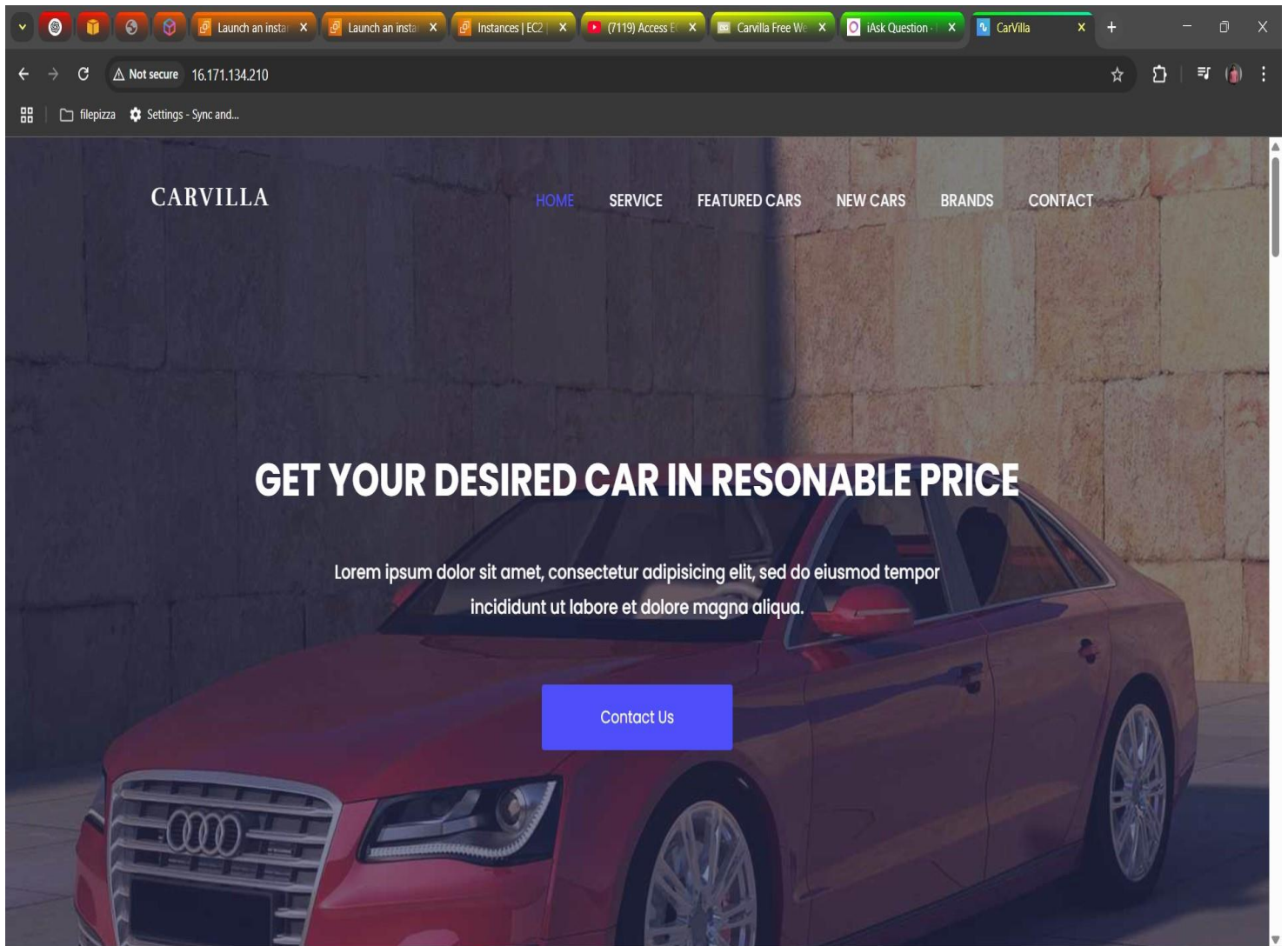
Below the script is a checkbox labeled 'User data has already been base64 encoded'. The 'Summary' section on the right displays the following configuration:

- Number of instances: 1
- Software Image (AMI): Amazon Linux 2023 AMI 2023.7.2...read more (ami-0274f4b62b6ae3bd5)
- Virtual server type (instance type): t3.micro
- Firewall (security group): New security group
- Storage (volumes): 1 volume(s) - 8 GiB

A 'Free tier' notification is shown: 'Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage for t3.micro where t2.micro isn't'. At the bottom right, there are 'Cancel' and 'Launch instance' buttons, along with a 'Preview code' link.

//open scripted-web-hosting and go to detail and copy ipv4 public ip address

// after that hit the ipv4 public ip address <http://16.171.134.210/>

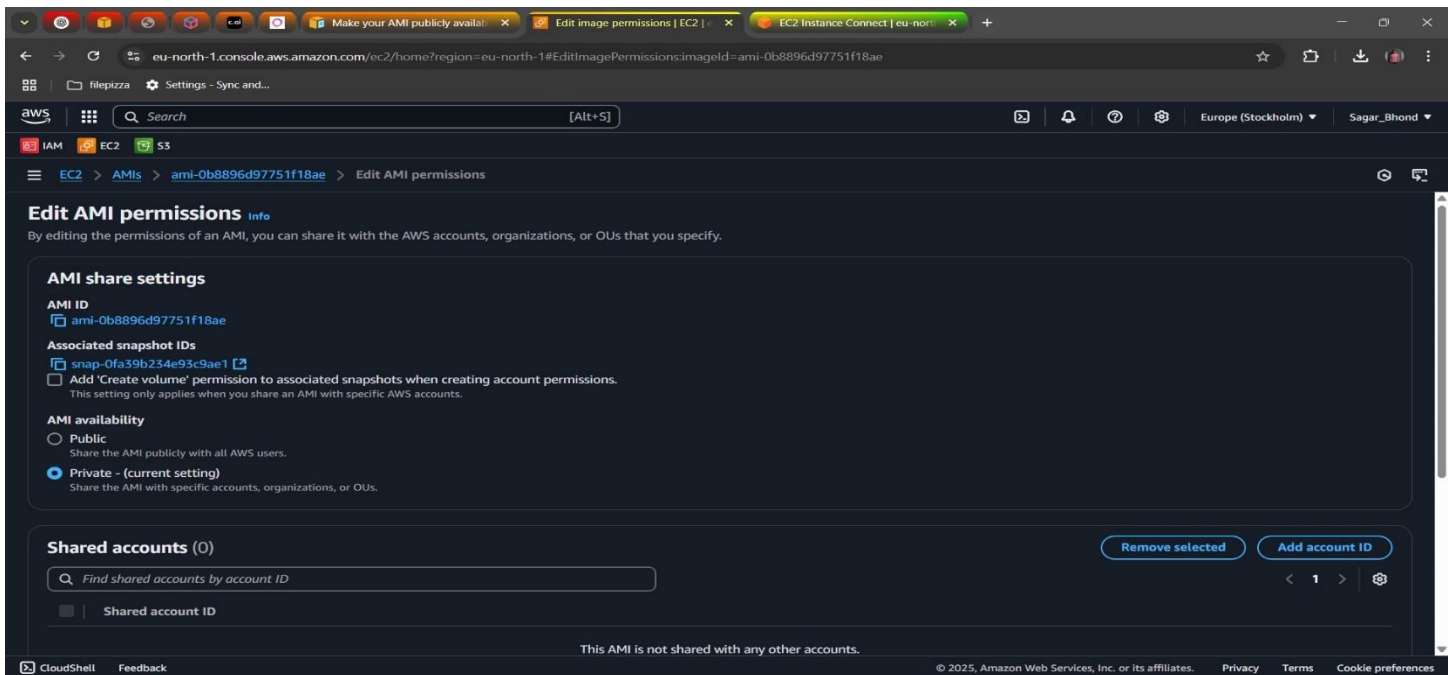


Task No:7

Name: sagar

ec2 :- Unblock Public Access for AMIs

// select ami -> Edit AMI permissions it shows disable public option.



// by using this command it display block-new-sharing or unblocked.

```
$ aws ec2 get-image-block-public-access-state --region eu-north-1
```

//To disable block public access

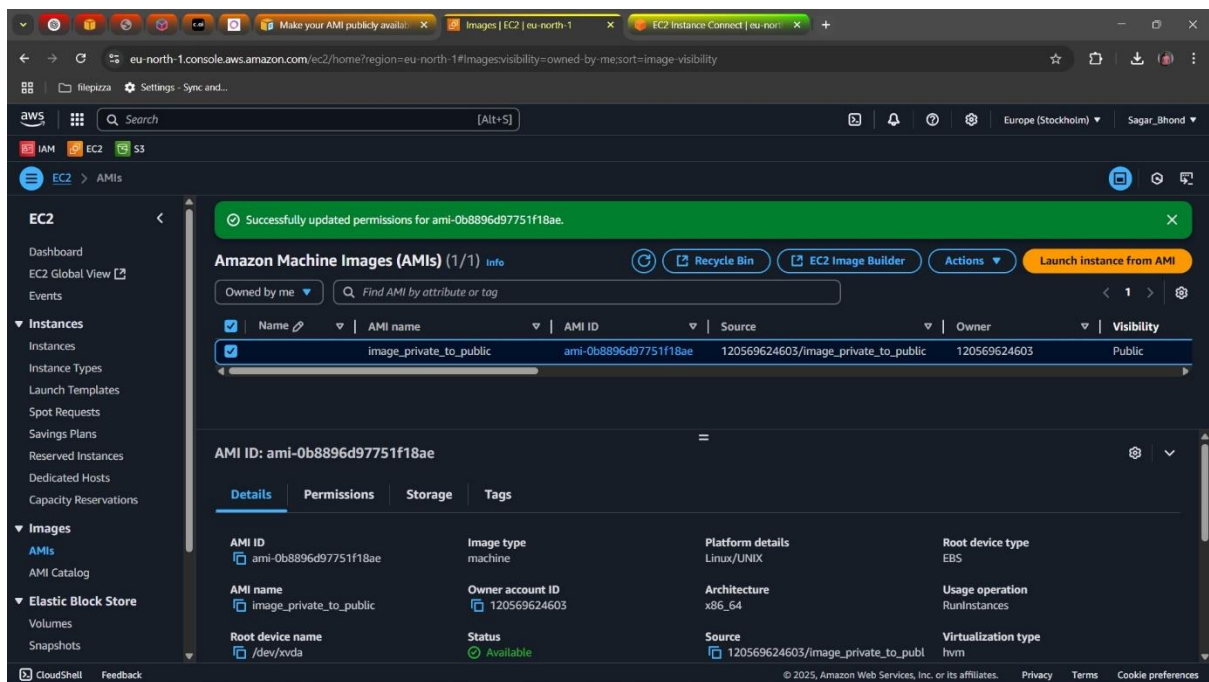
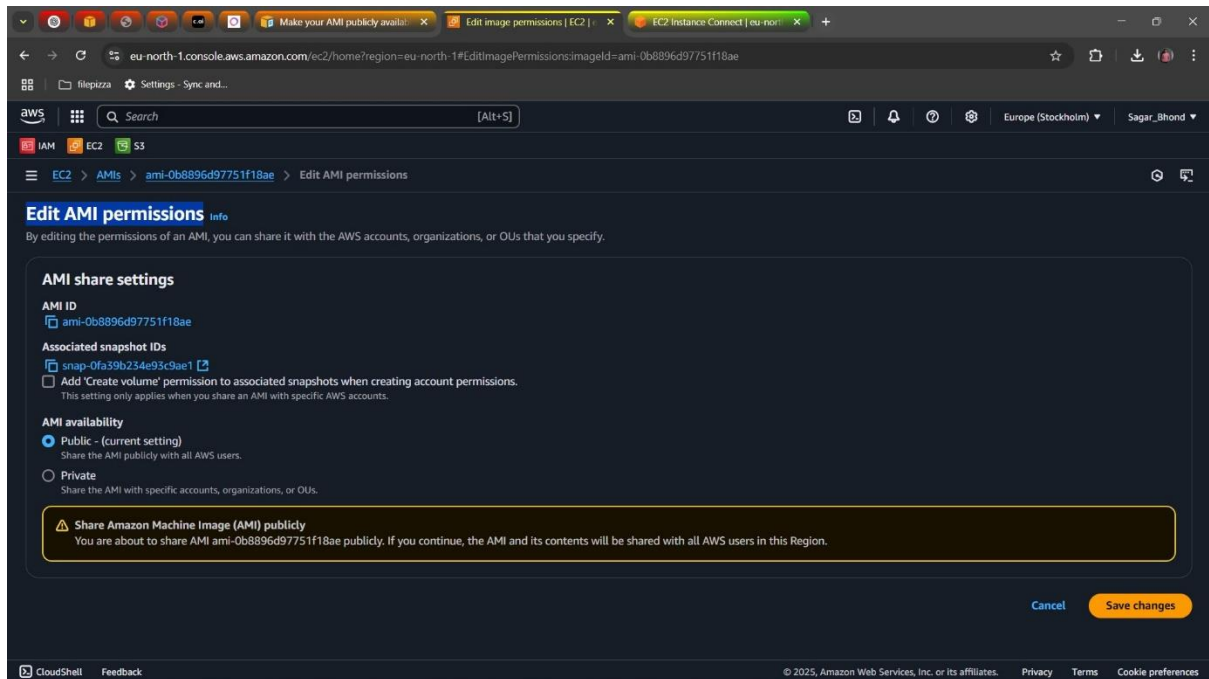
```
$ aws ec2 disable-image-block-public-access --region eu-north-1
```

//to see unblocked or not

```
$ aws ec2 get-image-block-public-access-state --region eu-north-1
```

```
[ec2-user@ip-172-31-32-133 ~]$ aws ec2 get-image-block-public-access-state --region eu-north-1
{
  "ImageBlockPublicAccessState": "block-new-sharing",
  "ManagedBy": "account"
}
[ec2-user@ip-172-31-32-133 ~]$ aws ec2 disable-image-block-public-access --region eu-north-1
{
  "ImageBlockPublicAccessState": "unblocked"
}
[ec2-user@ip-172-31-32-133 ~]$ aws ec2 get-image-block-public-access-state --region eu-north-1
{
  "ImageBlockPublicAccessState": "unblocked",
  "ManagedBy": "account"
}
[ec2-user@ip-172-31-32-133 ~]$
```

// after that enable public access it enable that option



Task No:8

Name: sagar

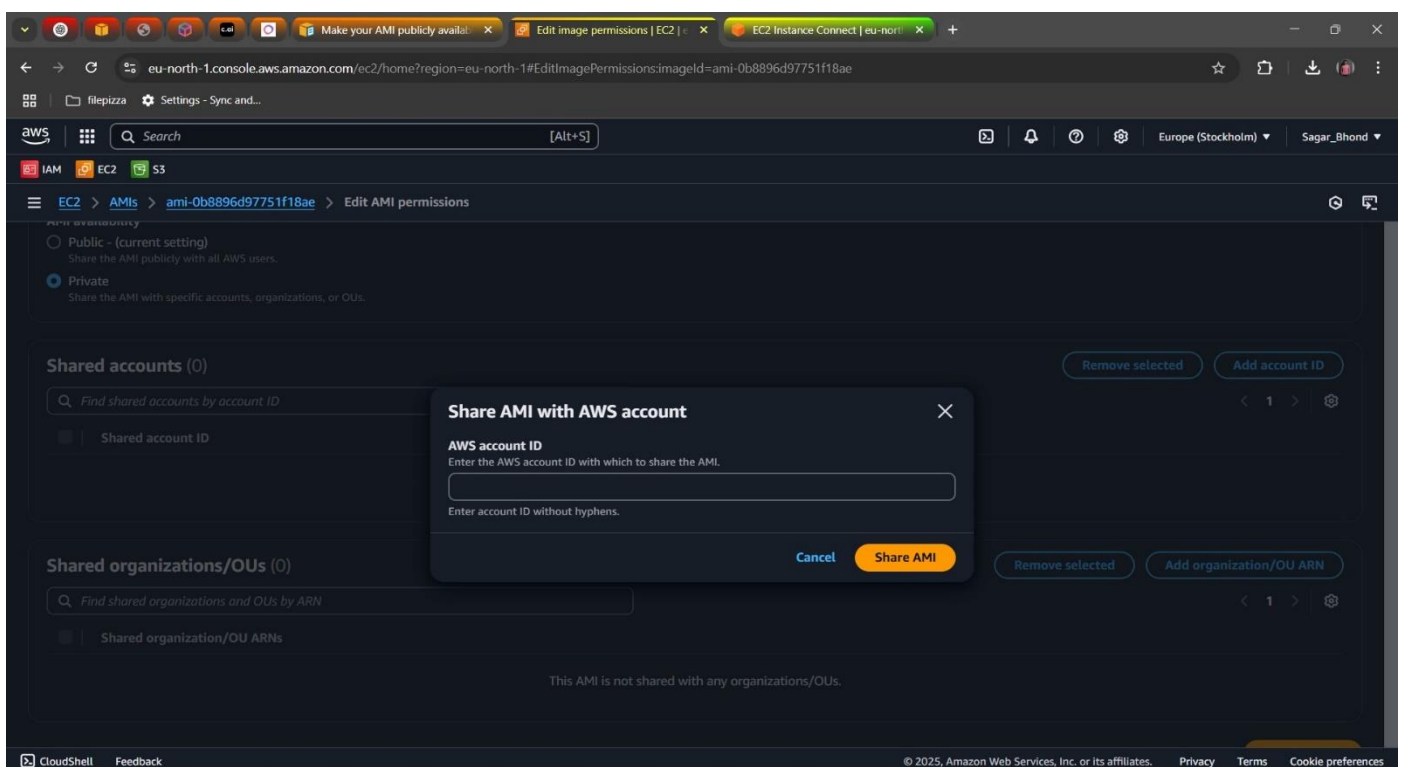
ec2 :- Share an EC2 AMI with a Friend's AWS Account.

// get your friend aws account id

//go to ec2 service -> ami -> select ami -> action -> Edit AMI permissions.

// AMI availability make private it gives option to share ami resource.

//past Shared accounts ID.



//after that it show This AMI is not shared with any organizations/OUs.

//enable it by using command

```
[ec2-user@ip-172-31-32-195 ~]$ aws configure
AWS Access Key ID [None]: AKIARYEUCPAN4HJ2TEEQ
AWS Secret Access Key [None]: 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
Default region name [None]:
Default output format [None]:
[ec2-user@ip-172-31-32-195 ~]$ aws ec2 describe-image-attribute --image-id ami-0abcdef1234567890 --attribute launchPermission
An error occurred (InvalidAMIID.Malformed) when calling the DescribeImageAttribute operation: The image ID 'ami-0abcdef1234567890' is malformed
[ec2-user@ip-172-31-32-195 ~]$ aws ec2 describe-image-attribute --image-id ami-0beaf9ea917bebd07 --attribute launchPermission
{
  "ImageId": "ami-0beaf9ea917bebd07",
  "LaunchPermissions": [
    {
      "UserId": "288761773173"
    }
  ]
}
[ec2-user@ip-172-31-32-195 ~]$ aws ec2 modify-image-attribute --image-id
```


//To share an AMI with an organization

```
[ec2-user@ip-172-31-32-195 ~]$ aws ec2 modify-image-attribute --image-id ami-0beaf9ea917bebd07 --launch-permission "Add=[(OrganizationArn=arn:aws:organizations::123456789012:organization/o-123example)]"
```

//after that check friends account ec2 ami resource whether ami is recived or not.

The screenshot shows the AWS Management Console for the 'eu-north-1' region. The main content area displays 'Amazon Machine Images (AMIs) (1) Info'. Below this, there is a table of private images. The table has columns for Name, AMI name, AMI ID, Source, Owner, and Vis. One image is listed with the name 'share_ami_to_friend' and AMI ID 'ami-0beaf9ea917bebd07'. The console also features a sidebar on the left with navigation options like Instances, Images, Elastic Block Store, and Network & Security.

Name	AMI name	AMI ID	Source	Owner	Vis
<input type="checkbox"/>	share_ami_to_friend	ami-0beaf9ea917bebd07	120569624603/share_ami_to_friend	120569624603	Priv

Task No:9

Name: sagar

ec2 :- EC2 recover loss key-pair .

```
// host web sit in our instance.
```

```
// first gen key after that go to git bash and next change dir to that key
```

```

sagar_c7otrfh@SagarBhond MINGW64 /C:/ProgramData/Microsoft/Windows/Start Menu/Programs/Git
$ ssh -i recover-key.pem ec2-user@13.201.83.76
The authenticity of host '13.201.83.76 (13.201.83.76)' can't be established.
ED25519 key fingerprint is SHA256:bHainIXT7Ngm/HX1T02pZUNKrny1dQZp682vxRl13Xs.
This host key is known by the following other names/addresses:
  ~/.ssh/known_hosts:1: ec2-13-201-83-76.ap-south-1.compute.amazonaws.com
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '13.201.83.76' (ED25519) to the list of known hosts.

#_
~  ##### Amazon Linux 2023
~  #####
~  #####
~  #####
~  #/
~  V~' -> https://aws.amazon.com/linux/amazon-linux-2023
~  .-.-
~  _/m/ ' -.-
Last login: Thu Apr 10 10:03:29 2025 from 13.233.177.3
[ec2-user@ip-172-31-11-200 ~]$ |

```

```
//after that logout the login exit
```

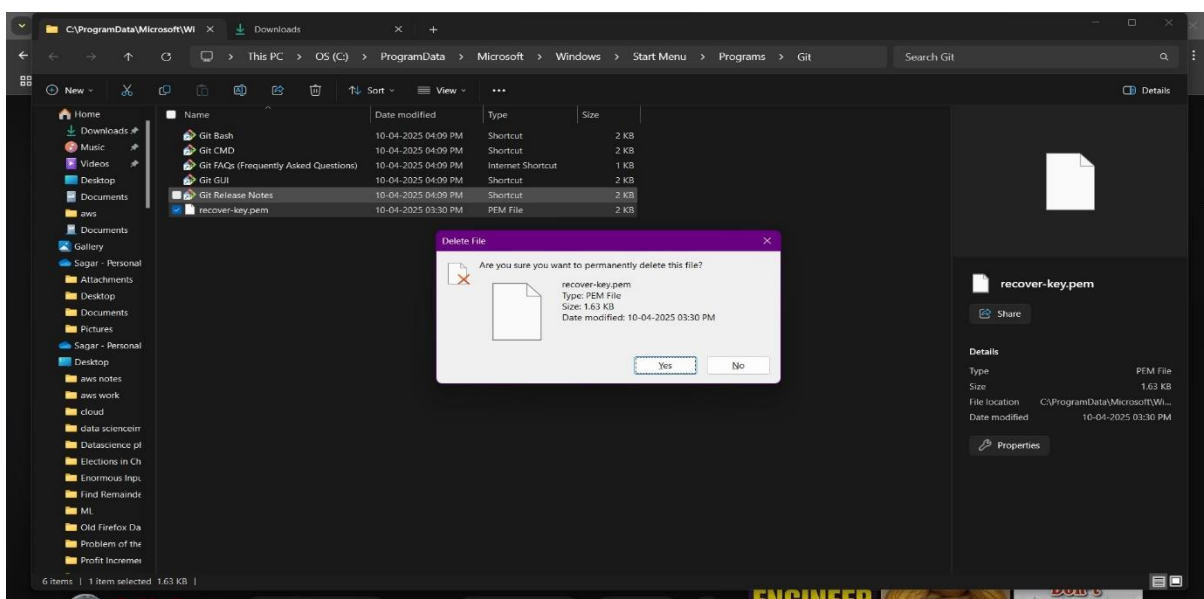
```
//after that delete key-pair permanently.
```

```
// in instance go to security -> security group link
```

```
// change inbound ssh custom to anywhere.
```

```
// after that connect your instance to server of aws.
```

```
// how to create new key-pair ,first of all delete old key-pair.
```




```
// create new key-file
```

```
[ec2-user@ip-172-31-11-200 ~]$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ec2-user/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
```

```
[ec2-user@ip-172-31-11-200 ~]$ cd .ssh/  
[ec2-user@ip-172-31-11-200 .ssh]$ ls  
authorized_keys id_rsa id_rsa.pub  
[ec2-user@ip-172-31-11-200 .ssh]$ cat id_rsa  
-----BEGIN OPENSSH PRIVATE KEY-----  
b3B1bnNzaC1rZXktZjEAAAAAG5vbmUAAAABm9uZQAAAAAAAAAABAABlWAAAdzc2gtcn.  
NhAAAAAwEAAQAAAYEAwbNrPwKt1lyoGTLXP5BkbbtCgrXl12VnxYlzim3xnHSB/Hr4z  
eOMMGYNme97ZlgGBKIDBGyVRBUI1Mj8Q1KL4L+LnZWrrllIoz6XOOxyWbj2eQB8tSiYKu8PK  
GCzumVgmWegSKKPHCGwpxy+a+jCBGNBo3wfepemYIbwblhq6f/wXxr2vNG6SyzxluhRvNT1+  
UJncn6q9Z3ctel5v9ja1vntjCBGNBo3wfepemYIbwblhq6f/wXxr2vNG6SyzxluhRvNT1+
```

```
// private key convert into new formate.
```

```
[ec2-user@ip-172-31-11-200 .ssh]$ ssh-keygen -p -m PEM -f id_rsa
Key has comment 'ec2-user@ip-172-31-11-200.ap-south-1.compute.internal'
Enter new passphrase (empty for no passphrase):
Enter same passphrase again:
```

```
//change private key format pest in vs code lflr into lrr formate and create private key in your device with .pem extention.
```

```
//change your ssh to anywhearpv-4.
```

```
// authorized key have there old public key override the new key-pair which authorize key
```

```
[ec2-user@ip-172-31-11-200 .ssh]$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDSwPxy56xp3PqVENf8J8d7YjxLNpJfcbWTEfg/2rDrJULjSPQa0BeANY9bnI4+67z4vxJ3ztO0aFH
jWtS4I4BractYTC/BVNxb9moivXzv02oM15ZvGL/HLyUimJRRs/bCTH4xJmmdkLeOEX7F5oAdvHywdznPZHjxCe0gJNvytDf8v0deBJeDHF3xesCgFsf
zFj7sjy6vxLsW4gTgjC134ckqAAH5Vb2DiinFuZ0jI3V recover-key
[ec2-user@ip-172-31-11-200 .ssh]$ cp id_rsa.pub authorized_keys
[ec2-user@ip-172-31-11-200 .ssh]$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGBuevNsiTvXI4lBmtc+xDkGrttMKCteXXZwdfIVokBdeefIH8evjN44wzpg2YT3tnWAYGQh0EzjA9
CZ26BKQo8ClpanHLSg+MIEY0Gjfb+16zghvBsewqnr9bFdhA83pLLF2W6FfidOX7U1YIN6HUXAda2qXmc2/7mvsO2MY6pc3/zWVv1PTFgePmq3mt1En
QyYfIqOqlG05DgYHqgiZRNnttoDS9m80d884Kmcw+7OX+wd1Z08PbtJ2NPKk07ngo/jHEMcbNRI+ZEMVbRY2osOr7+wg0qAcxEPFOYNgVDS9UVXwiH1
3+jGgozi/TKqFdryHx59pejBoUJ2G51Tkq0a+8quS0TV7NeMJU= ec2-user@ip-172-31-11-200 - ssh-agent -i
```

```
// login to your ec2 instance.
```

[illegible]

//check your hosted web site are display or not.

