

### 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 .  |
| 10    | Create Network Load Balancer  |
| 11    | Vpc: add multiple cidr in single vpc  |
| 12    | Vpc: can we delete main rout table  |
| 13    | Vpc: can we delete default vpc ,if delete how can you recover it  |
| 14    | Vpc: endpoints configuration  |
| 15    | Lambda: ec2 4 developer to give login and log out time automatically via code   |

---

## Task No:1

Name: sagar

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

//create policy

The screenshot shows the AWS Policy Generator interface. In Step 1: Select Policy Type, 'IAM Policy' is selected. In Step 2: Add Statement(s), the effect is set to 'Allow'. The AWS Service is 'Amazon EC2'. The condition 'aws:RequestedRegion' is set to 'ap-south-1'. The generated policy statement is:

| Effect | Action | Resource | Conditions   |
|--------|--------|----------|--|
| Deny   | *      | *        | StringNotEqualsIfExists<br>aws:RequestedRegion: "ap-south-1" |

In Step 3: Generate Policy, the policy is described as a container for statements. The 'Generate Policy' button is highlighted.

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.

Policies | IAM | Global    ec2-region-specific | IAM | Global

awspolicygen.s3.amazonaws.com/policygen.html

Amazon EC2

Actions -- Select Actions -- All Actions (\*)

Amazon Resource Name (ARN)

Policy JSON Document

You added the following statements:

Effect Deny

Effect Allow

Step 3: Review and Save

A policy is a 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": "Stmt17423826738005", "Action": "*", "Effect": "Deny", "Resource": "*"}, {"Condition": {"StringNotEqualsIfExists": {"aws:RequestedRegion": "ap-south-1"}}, "Action": "*", "Effect": "Allow", "Resource": "*"}]}
```

**Close**

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.

©2019, Amazon Web Services LLC or its affiliates. All rights reserved.

An [amazon.com](#) company

Roles | IAM | Global    sagar | IAM | Global    Policies | IAM | Global

us-east-1.console.aws.amazon.com/fam/home?region=eu-north-1#/policies?type=customer

filepizza Settings - Sync and...

aws Search [Alt+S]

IAM EC2 S3

IAM > Policies

Identity and Access Management (IAM)

Identity providers Account settings Root access management New

Access management User groups Users Roles Policies

Access reports Access Analyzer External access Unused access Analyzer settings Credential report

CloudShell Feedback

## Policies (1337) Info

A policy is an object in AWS that defines permissions.

Filter by Type

| Policy name         | Type             | Used as                | Description |
|---------------------|------------------|------------------------|-------------|
| ec2_write           | Customer managed | None                   |             |
| ec2-region-specific | Customer managed | Permissions policy (1) |             |

Actions Delete Create policy

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

Screenshot of the AWS EC2 Instances page in the Asia Pacific (Mumbai) region. The instance ID i-0654d686ecdf74f2 is selected. The instance is running, t2.micro, and initializing.

| Name | Instance ID        | Instance state | Instance type | Status check | Alarm status  | Availability Zone | Public IPv4 |
|------|--------------------|----------------|---------------|--------------|---------------|-------------------|-------------|
| new  | i-0654d686ecdf74f2 | Running        | t2.micro      | Initializing | View alarms + | ap-south-1a       | ec2-13-232  |

The sidebar shows navigation links for EC2, Instances, Images, and Elastic Block Store.

Screenshot of the AWS EC2 Instances page in the United States (N. Virginia) region. The instance ID i-0654d686ecdf74f2 is selected. A red error message indicates insufficient permissions:

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

The sidebar shows navigation links for EC2, Instances, Images, and Elastic Block Store.

## Task No:2

Name: sagar

\*AWS-MFA on S3 - Notepad

Step:1 \$ aws configure

Step:2 \$ aws s3api list-buckets

//check bucket list

```
sagar_c7otrfh@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

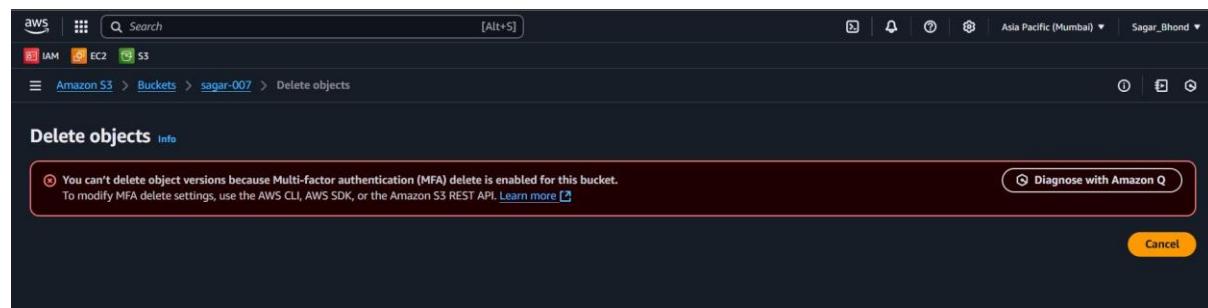
//check versioning is on or off

```
sagar_c7otrfh@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_c7otrfh>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" & "secrete 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 policey 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"><li>DateGreaterThan<ul style="list-style-type: none"><li>aws:CurrentTime: "2025-03-26T11:42:07+05:30"</li></ul></li><li>DateLessThanEquals<ul style="list-style-type: none"><li>aws:CurrentTime: "2025-03-26T12:02:08+05:30"</li></ul></li></ul> |

**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

The screenshot shows the IAM user 'sai' with the following details:

- ARN:** arn:aws:iam::120569624603:user/sai
- Created:** March 26, 2025, 11:27 (UTC+05:30)
- Console access:** Enabled without MFA
- Last console sign-in:** Today
- Access key 1:** Create access key

**Permissions** tab selected. **Permissions policies (1)**: EC2AutoStopPolicy (Customer managed, Attached via Directly). **Permissions boundary (not set)**.

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

//after 10 minute it automatically logout

The screenshot shows the IAM user's security credentials:

- Account details:** User name: sai, AWS account ID: 120569624603.
- AWS IAM credentials:** You have been signed out. A message states: "You've been signed out of the AWS console. AWS sessions in all tabs have been signed out." A yellow button says "Sign in again".
- Console sign-in:** Console sign-in link: https://120569624603.signin.aws.amazon.com/console
- Console password:** Access denied. Message: "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." Details: User: arn:aws:iam::120569624603:user/sai, Action: iam:GetLoginProfile, Context: no identity-based policy allows the action.

The screenshot shows the IAM users page:

- Identity and Access Management (IAM) sidebar:** Dashboard, Access management (User groups, Roles, Policies, Identity providers, Account settings, Root access management), Access reports (Analyzer, External access, Unused access, Analyzer settings, Credential report, Organization activity).
- Users (0) page:** An IAM user is an identity with long-term credentials that is used to interact with AWS in an account. A search bar and a table header: User name, Path, Group, Last activity, MFA, Password age, Console last sign-in, Acc.
- Access denied message:** 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. Details: 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. A "Diagnose with Amazon Q" button is present.



## 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  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 (\*)

Actions  Select 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  | <input checked="" type="radio"/> s3:GetObject | arn:aws:s3:::s3-logoutafter-3hr/sagar_Resume.pdf | <ul style="list-style-type: none"><li>DateGreaterThanOrEqualTo<ul style="list-style-type: none"><li>aws:CurrentTime: "2025-03-26T13:50:00+05:30"</li></ul></li><li>DateLessThanOrEqualTo<ul style="list-style-type: none"><li>aws:CurrentTime: "2025-03-26T16:50:00+05:30"</li></ul></li></ul> |

**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  Select 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.

**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-3hr/sagar_Resume.pdf", "Condition": { "DateGreaterThanOrEqualTo": { "aws:CurrentTime": "2025-03-26T13:50:00+05:30" }, "DateLessThanOrEqualTo": { "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

The screenshot shows the AWS IAM User Details page for a user named 's3-logoutafter-3hor'. The 'Permissions' tab is active, displaying a list of policies attached to the user. Two policies are listed: 'AmazonS3FullAccess' (AWS managed) and 's3\_policy\_logout\_after\_20min' (Customer managed). The 's3\_policy\_logout\_after\_20min' policy is highlighted.

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

//uploade your resume on their

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

The screenshot shows the AWS S3 Objects page for a bucket named 's3-logoutafter-3hor'. A single object, 'sagar\_Resume.pdf', is listed. The 'Actions' menu for this object is open, showing options such as 'Share with a presigned URL', 'Copy', 'Move', and 'Delete'.

//set time of 3 hour

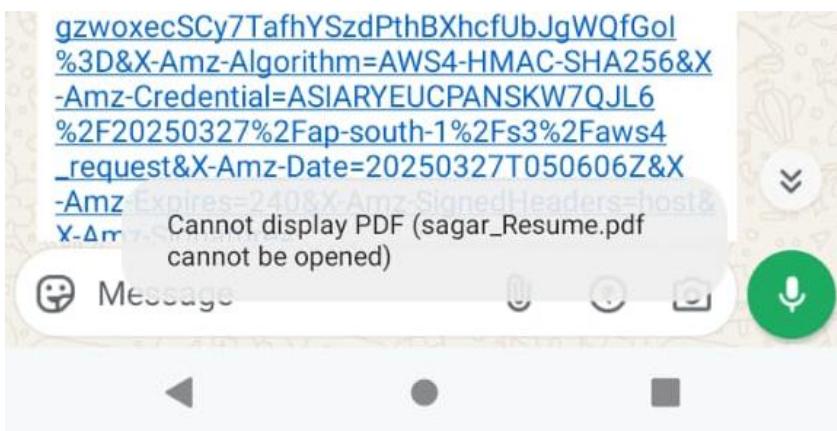
The screenshot shows the 'Share with a presigned URL' dialog box. It includes a warning message: 'Presigned URLs are used to grant access to an object for a limited time. Learn more'. Below this is a note: 'Anyone can access the object with this presigned URL until it expires, even if the bucket, and object are private.' A section titled 'Time interval until the presigned URL expires' explains that URLs can expire up to 12 hours or until session expires. The 'Number of hours' input field is set to '3'. At the bottom, a note states: 'After you create the presigned URL, it's automatically copied to your clipboard.'

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

The screenshot shows the AWS S3 console for a bucket named "s3-logoutafter-3hor". A green banner at the top indicates a presigned URL has been created and copied to the clipboard. The "Objects" tab is selected, displaying one object: "sagar\_Resume.pdf" (Type: pdf). The object was last modified on March 26, 2025, at 16:08:22 (UTC+05:30), with a size of 148.5 KB and a storage class of Standard. The object is selected, as indicated by a checkmark icon.

| Name             | Type | Last modified                           | Size     | Storage class |
|------------------|------|---|----------|---------------|
| sagar_Resume.pdf | pdf  | March 26, 2025, 16:08:22<br>(UTC+05:30) | 148.5 KB | Standard      |

//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

**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  | <input type="checkbox"/> s3>ListBucket                                       | arn:aws:s3:::sagar-009   | <i>None</i> |
| Allow  | <input type="checkbox"/> s3GetObject<br><input type="checkbox"/> s3PutObject | arn:aws:s3:::sagar-009/* | <i>None</i> |

**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.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "Stmt1743094718987",  
            "Action": [  
                "s3>ListBucket"  
            ],  
            "Effect": "Allow",  
            "Resource": "arn:aws:s3:::sagar-009"  
        },  
        {  
            "Sid": "Stmt1743094779514",  
            "Action": [  
                "s3GetObject",  
                "s3PutObject"  
            ],  
            "Effect": "Allow",  
            "Resource": "arn:aws:s3:::sagar-009/*"  
        }  
    ]  
}
```

Step:3 give access to only s3\_specific\_obj\_grant\_perm\_read\_write\_list.

The screenshot shows the AWS IAM User details page for the user 's3\_specific\_obj\_grant\_perm\_read\_write\_list'. The 'Permissions' tab is selected, displaying two attached policies: 'AmazonS3FullAccess' (AWS managed) and 's3\_specific\_obj\_grant\_perm\_read\_write\_list' (Customer managed). The 's3\_specific\_obj\_grant\_perm\_read\_write\_list' policy is highlighted.

| Policy name                                | Type             | Attached via |
|--|------------------|--------------|
| AmazonS3FullAccess                         | AWS managed      | Directly     |
| s3_specific_obj_grant_perm_read_write_list | Customer managed | Directly     |

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

The screenshot shows the AWS S3 General purpose buckets page. It lists three buckets: 's3-logoutafter-3hor', 'sagar-007', and 'sagar-009'. Each bucket row includes links to 'View analyzer for ap-south-1' and its creation date.

| Name                | AWS Region                       | IAM Access Analyzer          | Creation date                        |
|---------------------|----------------------------------|------------------------------|--------------------------------------|
| s3-logoutafter-3hor | Asia Pacific (Mumbai) ap-south-1 | View analyzer for ap-south-1 | March 26, 2025, 12:01:17 (UTC+05:30) |
| sagar-007           | Asia Pacific (Mumbai) ap-south-1 | View analyzer for ap-south-1 | March 24, 2025, 10:43:02 (UTC+05:30) |
| sagar-009           | Asia Pacific (Mumbai) ap-south-1 | View analyzer for ap-south-1 | March 27, 2025, 22:18:41 (UTC+05:30) |

//after I add only s3\_spacific\_obj\_grant\_perm\_read\_write\_list it look like.

The screenshot shows the AWS S3 console with the 'Directory buckets' tab selected. A single bucket named 'Error' is listed, with its status shown as 'Access Denied'. A red box highlights this entry. The URL in the address bar is <https://ap-south-1.console.aws.amazon.com/s3/buckets?region=ap-south-1&bucketType=directory>.

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

The screenshot shows a web browser with two tabs open. The first tab is 'S3 buckets | S3 | eu-north-1'. The second tab is active and shows the URL <https://eu-north-1.console.aws.amazon.com/s3/buckets/sagar-009>. The browser interface includes an Incognito mode button.

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

The screenshot shows the AWS S3 console with the 'sagar-009' bucket selected. One object named 'task/' is listed under the 'Objects' tab. The URL in the address bar is <https://eu-north-1.console.aws.amazon.com/s3/buckets/sagar-009>.

AWS Search [Alt+S] Asia Pacific (Mumbai) wiper @ 1205-6962-4603

Amazon S3 Buckets sagar-009 task/ Copy S3 URI

task/

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

Objects (4) Copy S3 URI Copy URL Download Open Delete Actions Create folder Upload

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Find objects by prefix Show versions < 1 >

| Name          | Type | Last modified                        | Size     | Storage class |
|---------------|------|--------------------------------------|----------|---------------|
| Task No 1.pdf | pdf  | March 27, 2025, 23:15:19 (UTC+05:30) | 334.2 KB | Standard      |
| Task No 4.pdf | pdf  | March 27, 2025, 23:15:22 (UTC+05:30) | 518.3 KB | Standard      |
| Task No_3.pdf | pdf  | March 27, 2025, 23:15:25 (UTC+05:30) | 425.4 KB | Standard      |
| Task_No_2.pdf | pdf  | March 27, 2025, 23:15:26 (UTC+05:30) | 144.0 KB | Standard      |

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 CloudFormation 'Launch an instance' wizard. The current step is 'User data - optional'. The user data script is pasted into the text area:

```
#!/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/carvilla.zip
sudo unzip carvilla.zip
sudo mv carvilla-v1.0/* /var/www/html
```

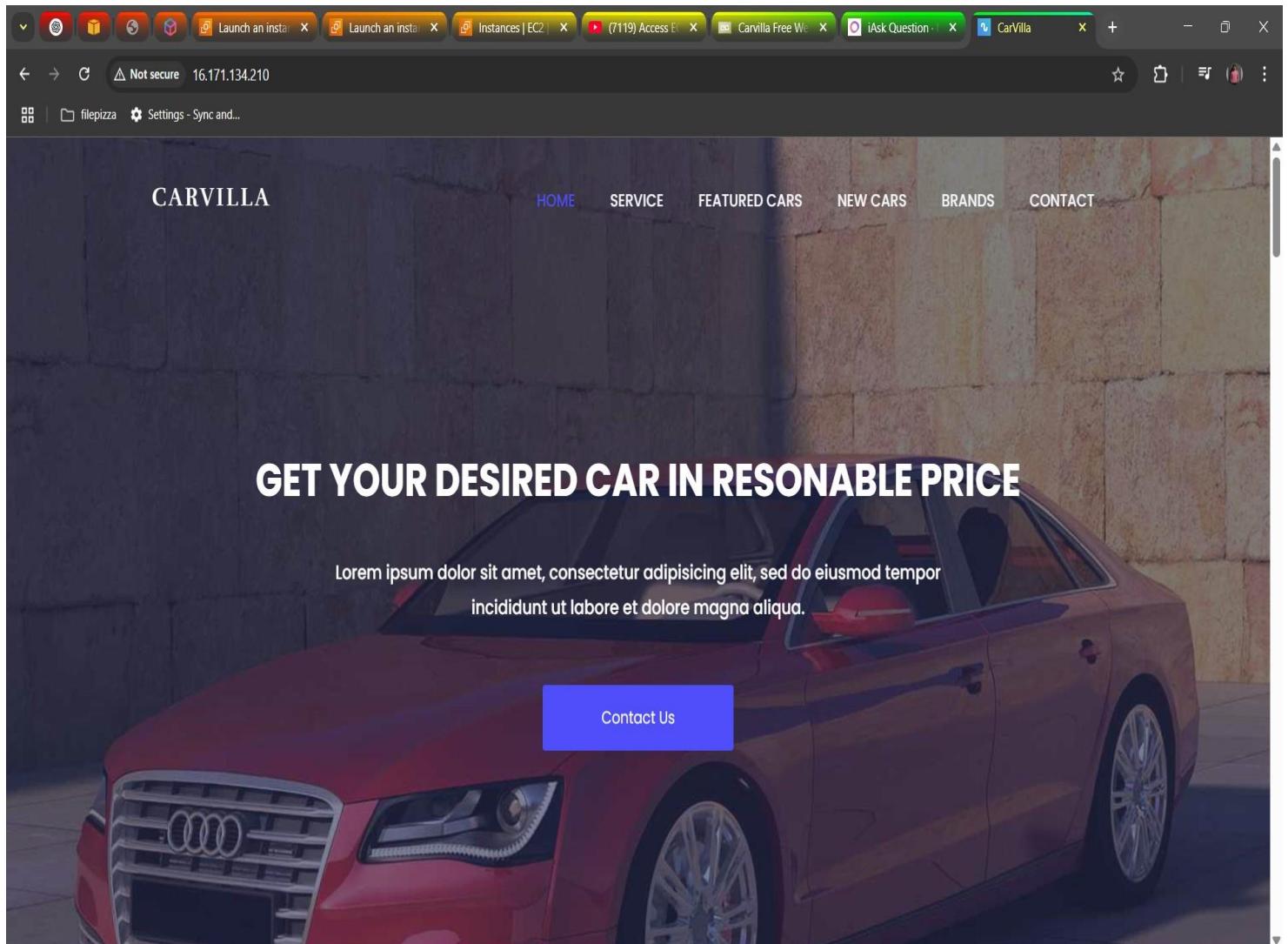
Below the script, there is a checkbox labeled "User data has already been base64 encoded". On the right side, the "Summary" section shows the following configuration:

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

A tooltip in the summary section states: "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 available."

At the bottom right, there are "Cancel", "Launch instance", and "Preview code" buttons.

//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.

The screenshot shows the AWS Management Console with the URL <https://eu-north-1.console.aws.amazon.com/ec2/home?region=eu-north-1#EditImagePermissions:imageId=ami-0b8896d97751f18ae>. The browser tab is titled 'Edit image permissions | EC2'. The main content area is titled 'Edit AMI permissions' for AMI ID 'ami-0b8896d97751f18ae'. Under 'AMI share settings', 'AMI ID' is 'ami-0b8896d97751f18ae' and 'Associated snapshot IDs' includes 'snap-0fa59b234e93c9ae1'. The 'AMI availability' section has two options: 'Public' (unchecked) and 'Private - (current setting)' (checked). Below this, the 'Shared accounts (0)' section shows a table with columns 'Find shared accounts by account ID', 'Remove selected', 'Add account ID', and pagination controls. A note at the bottom states 'This AMI is not shared with any other accounts.'

// 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

The screenshot shows the 'Edit AMI permissions' page in the AWS Management Console. The 'AMI share settings' section is visible, containing fields for 'AMI ID' (ami-0b8896d97751f18ae), 'Associated snapshot IDs' (snap-0fa39b254e93c9e1), and sharing options. A prominent yellow warning box at the bottom states: '⚠ Share Amazon Machine Image (AMI) publicly You are about to share AMI ami-0b8896d97751f18ae publicly. If you continue, the AMI and its contents will be shared with all AWS users in this Region.' Below the warning are 'Cancel' and 'Save changes' buttons.

The screenshot shows the 'Amazon Machine Images (AMIs)' list page. A green success message at the top states: 'Successfully updated permissions for ami-0b8896d97751f18ae.' The table lists one AMI entry: 'image\_private\_to\_public' (ami-0b8896d97751f18ae). The 'Visibility' column shows 'Public'. The 'Actions' dropdown menu for this item includes 'Recycle Bin', 'EC2 Image Builder', and 'Launch instance from AMI'. The left sidebar shows navigation links for EC2, Instances, Images, and Elastic Block Store.

## Task No:8

Name: sagar

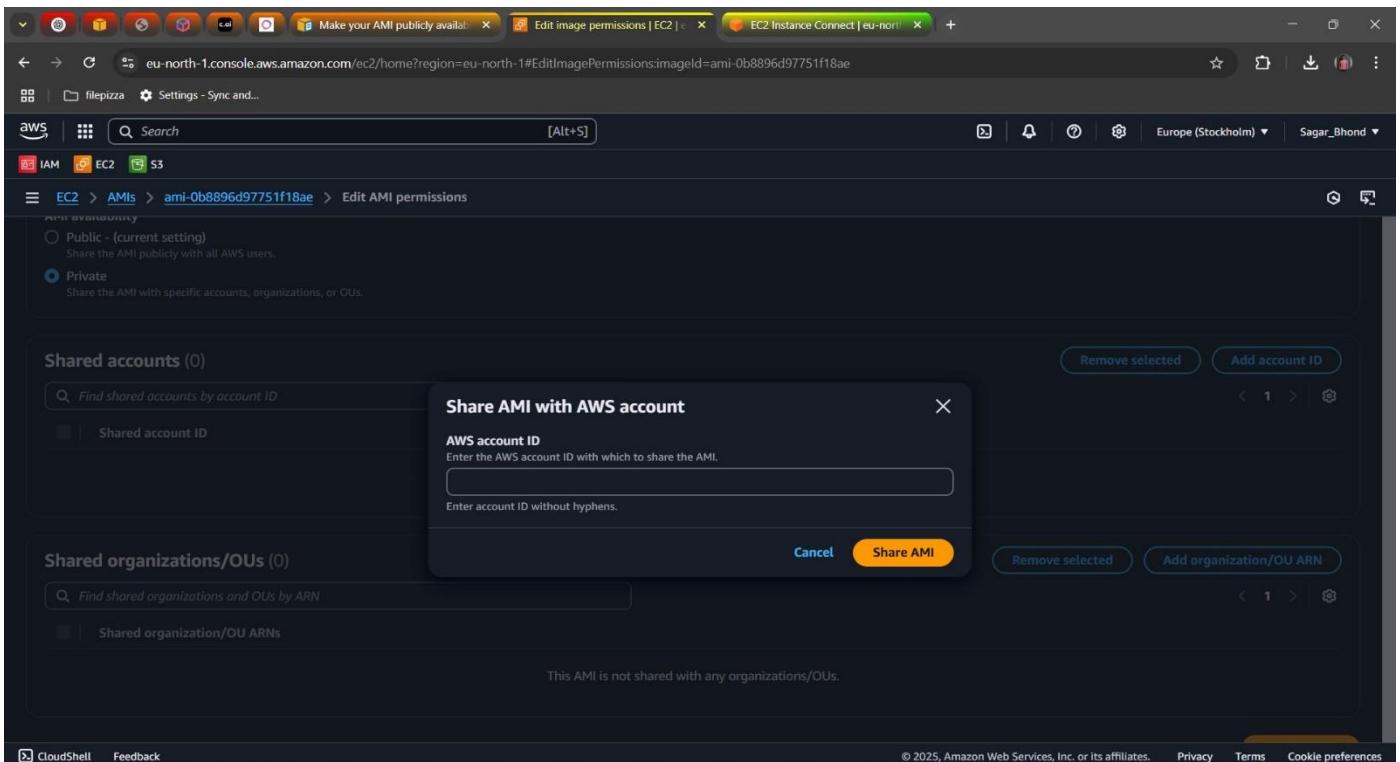
ec2 :- Share an EC2 AMI with a Friend's AWS Account.

// 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]: AKIARVEUCPAN4HJ2TREQ
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}]"
[ec2-user@ip-172-31-32-195 ~]$
```

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

Amazon Machine Images (AMIs) (1) [Info](#)

| Name                | AMI ID                | Source                           | Owner        | Visibility |
|---------------------|-----------------------|----------------------------------|--------------|------------|
| share_ami_to_friend | ami-0beaf9ea917bebd07 | 120569624603/share_ami_to_friend | 120569624603 | Private    |

Select an AMI

**AMIs**

- AMI Catalog

**Elastic Block Store**

- Volumes
- Snapshots
- Lifecycle Manager

**Network & Security**

- Security Groups

## 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/HX1T02pZUNKrnY1dQZp682vxR113Xs.
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
  ~\_ #####|      https://aws.amazon.com/linux/amazon-linux-2023
  ~\_ #####/
  ~\_ #####V~`_,`->
  ~\_ #####/
  ~\_ #####/
  ~\_ #####/
  ~\_ #####/
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

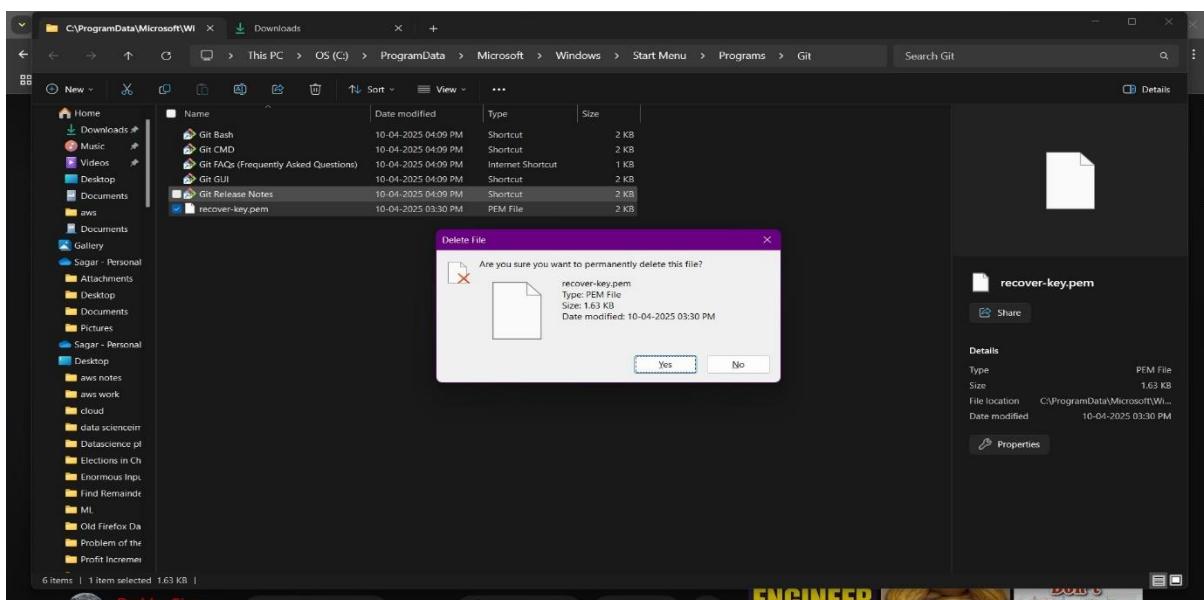
//after that delete key-pair permanently.

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

// change inbound ssh custom to anyware.

// 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):
```

// 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 lr formate and create private key in your device with .pem extention.

//change your ssh to anywhereipv-4.

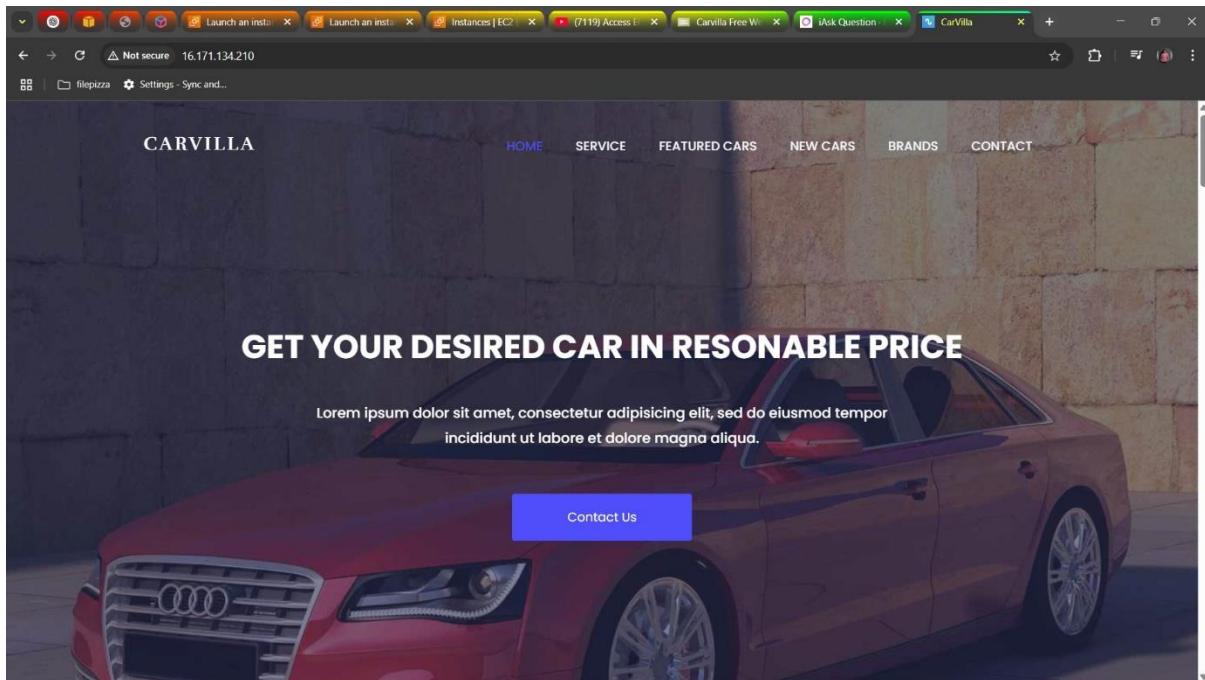
// authorized key have there old public key override the new key-pair whith authorize key

```
[ec2-user@ip-172-31-11-200 .ssh]$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQbQDswPxY56xp3PqvvnF8J8d7YjxLNpjfcBwTEfq/2rDJLjsPQAoBeANY9bnI4+67z4vxJ3zQt00aFH
5nGw14BractYC/BVNvX9moiVxZv0oM1zVgqL/H1YUimJRRs/btCTH4xJ5mjdKleOEox7F5oAdVHywdznPZhjxCeOgJNvylDf8voDebJeDHF3xesCgFs
zFj7Sjy6vxjLsw4tgjC1w3ckqWAHH5vb2DiiFnUz0J1V 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 AAAAB3NzaC1yc2EAAAQABAAQbQDswNxTiX14B1MtC+xDkGrTtMKCteXXZwDf1vOkbdeeFIH8evjN44wzpg2YT3tnWYQgh0EZjA9
ZCZ6BkQo8cLpanHL9g+MIEY0GjFb+16ZghvBseWqn9bfFdHa83pLlF2W6FFid0X7U1YING6UXAda2qXmc2/7Mvs02Y6pc3/zWVviPTFqeFmq3mtiEn
QyYrfi0qlq0g5yHqgqizRnnntoDs9M8oD884KmcW7OX+wD1208Pbtj2NPkk07ng/jHEmcBnRI+zEMVbRy2osor7-wq0qAcxFPF0YNgVds9UVXwiH1
3+jGgozi/TkqFdryHx59pejBouJ2G51Tkq0a+8quStv7NeMJU= ec2-user@ip-172-31-11-200.ap-south-1.compute.internal
```

// login to your ec2 instance.

```
sagar_c7otrfh@sagarbhond MINGW64 /C/ProgramData/Microsoft/Windows/Start Menu/Programs/Git
$ ssh -i new-privkey.pem ec2-user@13.201.83.76
,           #
~\_\_ ##### Amazon Linux 2023
~~ \_\_#####\
~~ \###|
~~      \#/   https://aws.amazon.com/linux/amazon-linux-2023
~~          V~'-->
~~          /
~~.._.-/ \
~/m/ .-/
Last login: Thu Apr 10 11:27:55 2025 from 13.233.177.5
[ec2-user@ip-172-31-11-200 ~]$ |
```

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



## Task No:10

Name: sagar

ec2 :- Create Network Load Balancer .

step 1 :- create instance

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with options like Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, and AMI Catalog. The main area displays a table titled 'Instances (6/10) Info' with columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4. All instances are listed as 'Running'. Below the table, it says '6 instances selected'. At the bottom, there are monitoring and CloudWatch agent configuration options.

step 2 :- create target group

The screenshot shows the AWS EC2 Target groups page. The left sidebar includes Images, Elastic Block Store, Network & Security, Load Balancing (with Target Groups selected), and Auto Scaling. The main content area shows a table titled 'Target groups (3) Info' with columns: Name, ARN, Port, Protocol, Target type, Load balancer, and VPC ID. The target types are all 'Instance'. At the bottom, it says '0 target groups selected' and 'Select a target group above.'

### Step 3 :- Create Load Balancer

The screenshot shows the AWS EC2 Load Balancers console. On the left, a navigation sidebar lists various services: Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces, Load Balancing (Load Balancers selected), Target Groups, Trust Stores, Auto Scaling, and Auto Scaling Groups. The 'Load Balancers' section is currently selected. The main pane displays a table titled 'Load balancers (1)'. The table has columns for Name, DNS name, State, VPC ID, Availability Zones, Type, and Date created. One row is listed: 'network-lb' with a DNS name of 'network-lb-b0851afb03a4004.elb.eu-north-1.amazonaws.com', an Active state, VPC ID 'vpc-0f3a8d202235dbe9b', 3 Availability Zones, a network type, and a creation date of April 21, 2025. Below the table, a message says '0 load balancers selected' and 'Select a load balancer above.' The top right of the page has a 'Create load balancer' button. The bottom right of the page includes copyright information: '© 2025, Amazon Web Services, Inc. or its affiliates.', 'Privacy', 'Terms', and 'Cookie preferences'.

Step 4:- copy DNS and hit

The screenshot shows a web browser window with the URL 'network-lb-b0851afb03a4004.elb.eu-north-1.amazonaws.com'. The browser interface includes standard navigation buttons (back, forward, search, etc.) and a status bar indicating 'Not secure'. The main content area of the browser is blank, showing only the copied DNS address.

**network lb ip-172-31-17-2.eu-north-1.compute.internal**

## Task No:11

Name: sagar

Vpc: add multiple cidr in single vpc

The screenshot shows the AWS VPC console interface. The top navigation bar includes tabs for IAM, EC2, S3, VPC, Aurora and RDS, and DynamoDB. The main content area is titled "Adding multiple CIDRs VPC". The left sidebar shows "VPC | eu-north-1" and "eu-north-1.console.aws.amazon.com/vpcconsole/home?region=eu-north-1#EditVpcCidr;VpcId=vpc-082ec2ddff61106b1". The main content is divided into two sections:

- IPv4 CIDRs**: Shows two CIDR blocks:

| CIDR        | Status     |
|-------------|------------|
| 10.0.0.0/16 | Associated |
| 10.1.0.0/16 | Associated |

Each row has a "Remove" button.
- IPv6 CIDRs**: Shows a message: "You have no IPv6 CIDR blocks associated with your VPC." A "Add new IPv6 CIDR" button is present.

A sidebar on the right is titled "IPv4 CIDR block" and contains the following text:

Specify an IPv4 CIDR block (or IP address range) for your VPC.  
If there is an Amazon VPC IP Address Manager (IPAM) IPv4 address pool available in this Region, you can get a CIDR from an IPAM pool. If you select an IPAM pool, the size of the CIDR is limited by the allocation rules on the IPAM pool (allowed minimum, allowed maximum, and default).  
If there is no IPv4 IPAM pool in this Region, you can manually input an IPv4 CIDR. The CIDR block size must have a size between /16 and /28.

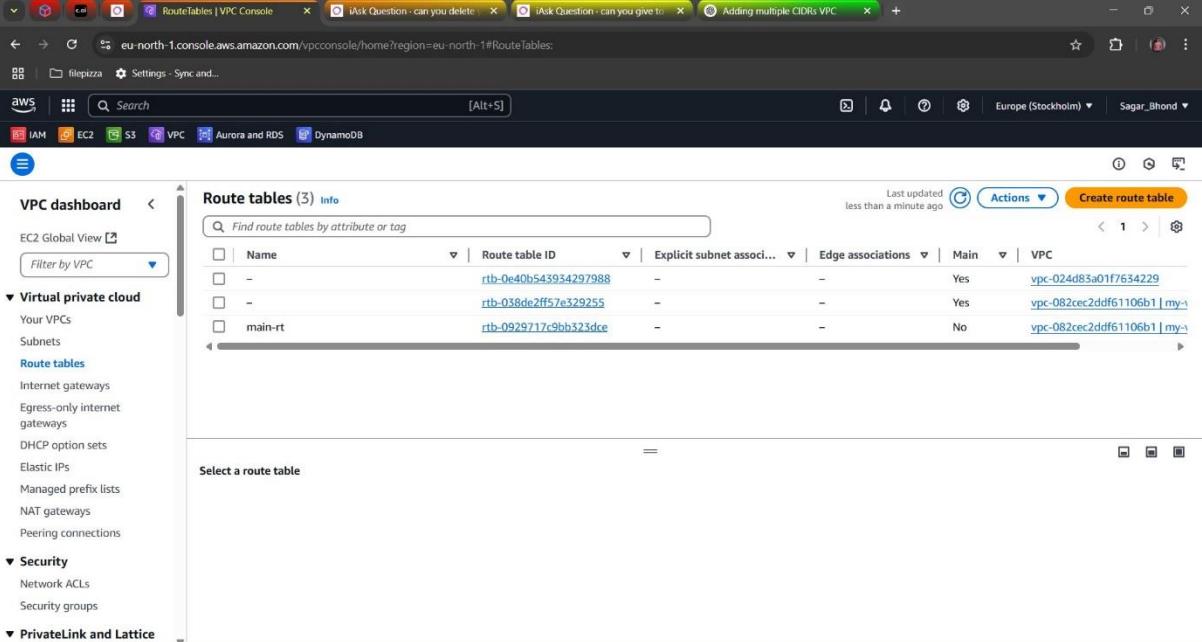
Learn more [\[i\]](#)  
[Edit VPC CIDRs](#)

## Task No:12

Name: sagar

Vpc: can we delete main rout table

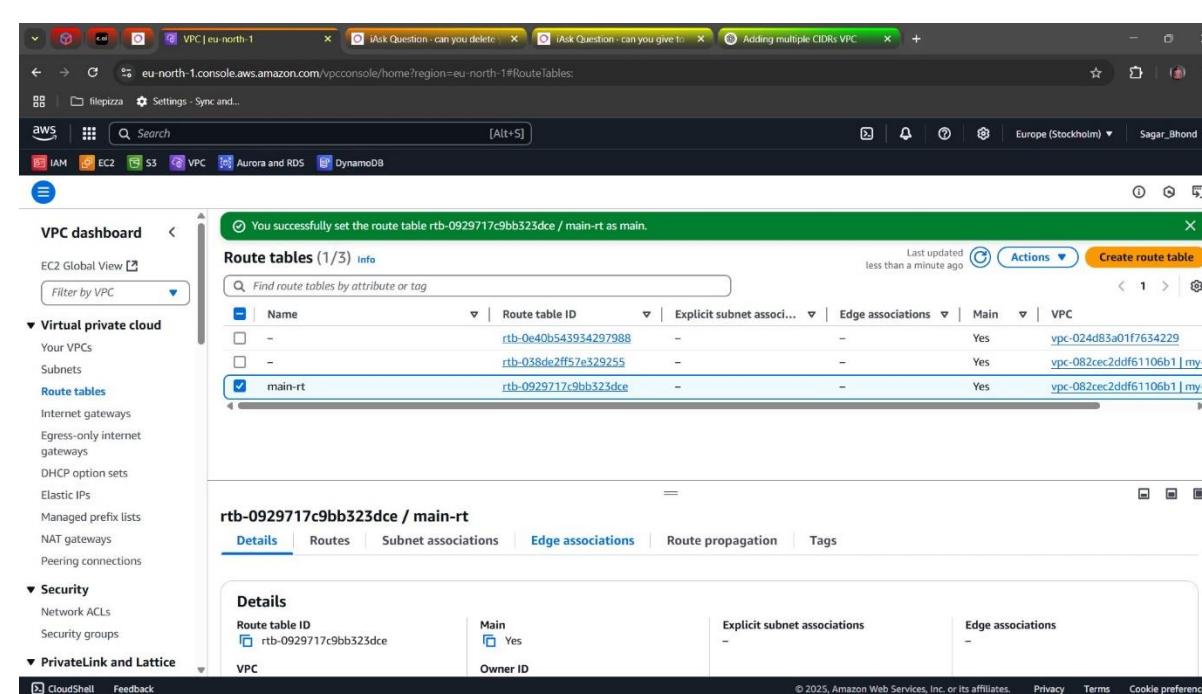
//if you wan to delete main rout table ,first you want to create new rout table and edit it as main rout table



The screenshot shows the AWS VPC Route Tables page. On the left, there's a sidebar with options like EC2 Global View, Virtual private cloud (Your VPCs, Subnets, Route tables), Security (Network ACLs, Security groups), and PrivateLink and Lattice. The main area displays a table of route tables:

| Name    | Route table ID        | Explicit subnet associ... | Edge associations | Main | VPC                           |
|---------|-----------------------|---------------------------|-------------------|------|-------------------------------|
| -       | rtb-0e40b543934297988 | -                         | -                 | Yes  | vpc-024dB3a01f7634229         |
| -       | rtb-038de2ff57e529255 | -                         | -                 | Yes  | vpc-082cec2ddff61106b1   my-v |
| main-rt | rtb-0929717c9bb323dce | -                         | -                 | No   | vpc-082cec2ddff61106b1   my-v |

Below the table, a message says "Select a route table".

The screenshot shows the same VPC Route Tables page after setting the 'main-rt' route table as the main route table. A green success message at the top says "You successfully set the route table rtb-0929717c9bb323dce / main-rt as main." The 'main-rt' row in the table now has a checked checkbox next to it. The table header includes a "Main" column with a dropdown menu.

Below the table, the details for the 'main-rt' route table are shown. The "Details" tab is selected, displaying the following information:

| Route table ID        | Main | Explicit subnet associations | Edge associations |
|-----------------------|------|------------------------------|-------------------|
| rtb-0929717c9bb323dce | Yes  | -                            | -                 |

Other tabs available include Routes, Subnet associations, Edge associations, Route propagation, and Tags.

Screenshot of the AWS VPC console showing the Route tables page. A modal dialog is open for deleting the route table 'rtb-038de2ff57e329255'.

**Route tables (1/3) info**

| Name    | Route table ID        | Explicit subnet assoc... | Edge associations | Main | VPC                           |
|---------|-----------------------|--------------------------|-------------------|------|-------------------------------|
| -       | rtb-038de2ff57e329255 | -                        | -                 | No   | vpc-082cec2dd61106b1   my-vpc |
| main-rt | rtb-0929717c9bb323dce | -                        | -                 | Yes  | vpc-082cec2dd61106b1   my-vpc |
| -       | -                     | -                        | -                 | -    | vpc-024d83a01f7634229         |

**Delete route tables**

The following route tables will be deleted permanently and can't be recovered later.

| Name | Route table ID        | VPC ID               |
|------|-----------------------|----------------------|
| -    | rtb-038de2ff57e329255 | vpc-082cec2dd61106b1 |

To confirm deletion, type **delete** in the field:

**Details**

| Route table ID                | Main         | Explicit subnet associations | Edge associations |
|-------------------------------|--------------|------------------------------|-------------------|
| rtb-038de2ff57e329255         | No           | -                            | -                 |
| VPC                           | Owner ID     |                              |                   |
| vpc-082cec2dd61106b1   my-vpc | 692450379765 |                              |                   |

**Actions**

Cancel   Delete

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

## Task No:13

Name: sagar

Vpc: can we delete default vpc ,if delete how can you recover it.

// yes you can delete default vpc , but it can not be recoverable you want to create new default vpc

The screenshot shows the AWS VPC console interface. On the left, the 'VPC dashboard' sidebar lists various VPC-related services like EC2 Global View, Subnets, Route tables, Internet gateways, Egress-only internet gateways, DHCP option sets, Elastic IPs, Managed preflight lists, NAT gateways, and Peering connections. The main area displays a confirmation message: 'You successfully deleted vpc-024d83a01f763422'. Below this, a table titled 'Will also be deleted' lists 8 resources that will be permanently deleted: igw-0ba815d1594feb73, sg-0ff9f2f67db973b4f, sg-09fdce5243b89075e, sg-0aa0d6f4947e192ea, and sg-0db2eb51a4bb95a50. A warning message states: 'Warning: If you delete this default VPC, you can't launch instances in this Region unless you specify a subnet in another VPC or create a new default VPC.' At the bottom, a checkbox is checked with the text 'I acknowledge that I want to delete my default VPC.' and a field containing 'delete default vpc'. Buttons for 'Cancel' and 'Delete' are visible.

The screenshot shows the 'Create default VPC' wizard. The top navigation bar includes links for IAM, EC2, S3, VPC, Aurora and RDS, and DynamoDB. The main content area is titled 'Create default VPC' with a sub-link 'Info'. A descriptive text box explains that a default VPC enables launching Amazon EC2 resources without creating a VPC, mentioning the creation of a default subnet in each Availability Zone, an internet gateway, and a route table. At the bottom right are 'Cancel' and 'Create default VPC' buttons.

## Task No:14

Name: sagar

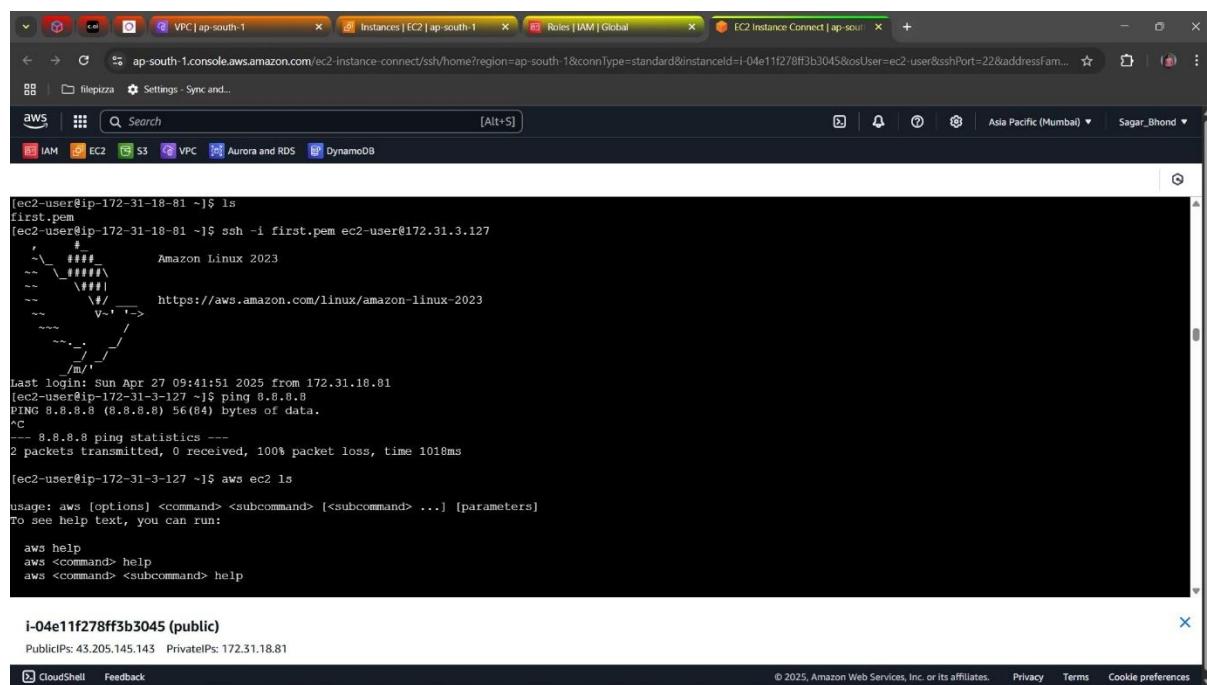
Vpc: endpoints configuration

//end point configuration is used specifically for connecting to Amazon S3,ec2,etc.

//create vpc in that vpc create public and private subnet ,and also create end points

//create public and private subnet instance

// accesing ec2 data from private instance.



The screenshot shows the AWS CloudShell interface with a terminal window open. The terminal output is as follows:

```
[ec2-user@ip-172-31-18-81 ~]$ ls
first.pem
[ec2-user@ip-172-31-18-81 ~]$ ssh -i first.pem ec2-user@172.31.3.127
Last login: Sun Apr 27 09:41:51 2025 from 172.31.18.81
[ec2-user@ip-172-31-3-127 ~]$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
^C
--- 8.8.8.8 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1018ms
[ec2-user@ip-172-31-3-127 ~]$ aws ec2 ls
usage: aws [options] <command> <subcommand> [<subcommand> ...] [parameters]
To see help text, you can run:
  aws help
  aws <command> help
  aws <command> <subcommand> help

i-04e11f278ff3b3045 (public)
PublicIPs: 43.205.145.143 PrivateIPs: 172.31.18.81
```

The terminal shows the user has logged in via SSH from a private instance (172.31.18.81) to a public instance (172.31.3.127). It then performs a ping to 8.8.8.8 and lists the AWS services available in the CloudShell.

## Task No:15

Name: sagar

Task :Lambda: ec2 4 developer to give login and log out time automatically via code

| Name    | Start Time | Stop Time |
|---------|------------|-----------|
| Ram     | 07:00      | 11:00     |
| Shyam   | 09:00      | 13:00     |
| Prem    | 14:00      | 18:00     |
| Krishna | 10:00      | 14:00     |

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with options like Dashboard, EC2 Global View, Events, Instances, Images, Elastic Block Store, and CloudShell. The main area is titled 'Instances (5) Info' and lists five instances:

| Name    | Instance ID         | Instance state | Instance type | Status check      | Alarm status  | Availability Zone | Public IPv4 D |
|---------|---------------------|----------------|---------------|-------------------|---------------|-------------------|---------------|
| my      | i-065dbdx4d1d1894e  | Terminated     | t5.micro      | -                 | View alarms + | eu-north-1b       | -             |
| shyam   | i-03612ab1e6165263f | Running        | t5.micro      | 3/3 checks passed | View alarms + | eu-north-1b       | ec2-56-228-25 |
| prem    | i-0487a721bab983798 | Running        | t5.micro      | 3/3 checks passed | View alarms + | eu-north-1b       | ec2-13-60-19  |
| ram     | i-055d5df4a7478d056 | Running        | t5.micro      | 3/3 checks passed | View alarms + | eu-north-1b       | ec2-13-51-165 |
| krishna | i-0a2e27290132716a8 | Running        | t5.micro      | 3/3 checks passed | View alarms + | eu-north-1b       | ec2-16-171-12 |

The screenshot shows the AWS IAM Roles page. On the left, there's a navigation sidebar with options like Identity and Access Management (IAM), Access management, and Access reports. The main area is titled 'Identity and Access Management (IAM)' and shows the 'Roles' section. A specific role, 'start-ec2', is selected. The 'Permissions boundary (not set)' section is visible at the bottom.

The 'AmazonEC2FullAccess' policy details are shown in a JSON editor:

```

1- {
2-   "Version": "2012-10-17",
3-   "Statement": [
4-     {
5-       "Action": "ec2:*",
6-       "Effect": "Allow",
7-       "Resource": "*"
8-     },
9-     {
10-       "Effect": "Allow",
11-       "Action": "elasticloadbalancing:*",
12-       "Resource": "*"
13-     },
14-     {
15-       "Effect": "Allow",
16-       "Action": "cloudwatch:*log",
17-       "Resource": "*"
18-     },
19-     {
20-       "Effect": "Allow",
21-       "Action": "logs:CreateLogGroup",
22-       "Resource": "*"
23-     }
24-   ]
25- }
  
```

```
stop-EC2-instances | Functions | start-ec2 | IAM | Global | Instances | EC2 | eu-north-1 | Start EC2 in 5 mins
```

```
lambda_function.py
```

```
import boto3
from datetime import datetime, timedelta

# AWS region
region = 'eu-north-1'
ec2 = boto3.client('ec2', region_name=region)

# Map each name to its instance ID and start/stop time (24-hour format)
instances = {
    'ram': {
        'id': 'i-055d5df4a7478d056',
        'start': 7,
        'stop': 11
    },
    'shyam': {
        'id': 'i-03612ab1e6165263f',
        'start': 9,
        'stop': 13
    },
    'prem': {
        'id': 'i-0487a721bab983798',
        'start': 14,
        'stop': 18
    },
    'krishna': {
        'id': 'i-0a2e27290132716a8',
        'start': 10,
        'stop': 14
    }
}

def lambda_handler(event, context):
    # Adjust UTC time to India Standard Time (UTC+5:30)
```

Ln 59, Col 1 Spaces: 4 UTF-8 CRLF Python Lambda Layout: US

```
stop-EC2-instances | Functions | start-ec2 | IAM | Global | Instances | EC2 | eu-north-1 | Start EC2 in 5 mins
```

```
lambda_function.py
```

```
def lambda_handler(event, context):
    # Adjust UTC time to India Standard Time (UTC+5:30)
    utc_now = datetime.utcnow()
    india_time = utc_now + timedelta(hours=5, minutes=30)
    current_hour = india_time.hour

    to_start = []
    to_stop = []

    for name, config in instances.items():
        if current_hour == config['start']:
            to_start.append(config['id'])
        elif current_hour == config['stop']:
            to_stop.append(config['id'])

    if to_start:
        ec2.start_instances(InstanceIds=to_start)
        print(f"Started instances: {to_start}")

    if to_stop:
        ec2.stop_instances(InstanceIds=to_stop)
        print(f"Stopped instances: {to_stop}")

    return {
        'statusCode': 200,
        'body': f"Checked at hour {current_hour}, started: {to_start}, stopped: {to_stop}"
    }
```

Ln 59, Col 1 Spaces: 4 UTF-8 CRLF Python Lambda Layout: US

```
stop-EC2-instances | Functions | start-ec2 | IAM | Global | Instances | EC2 | eu-north-1 | Start EC2 in 5 mins
```

```
lambda_function.py
```

```
20250513T162937 > http://webWorker > output_logging_20250513T162937 > 9-Execution Results.log
```

```
1 status: Succeeded
2 Test Event Name: my
3
4 Response:
5 {
6     "statusCode": 200,
7     "body": "Checked at hour 16, started: [], stopped: []"
8 }
9
10 Function Logs:
11 START RequestId: 247daafe-65f4-47d3-903e-0abf4a53366d Version: $LATEST
12 END RequestId: 247daafe-65f4-47d3-903e-0abf4a53366d
13 REPORT RequestId: 247daafe-65f4-47d3-903e-0abf4a53366d Duration: 5.47 ms Billed Duration: 6 ms Memory Size: 128 MB Max Memory Use
14
15 Request ID: 247daafe-65f4-47d3-903e-0abf4a53366d
```