

Name :- Samartha Santosh Lasure

Batch :- B-56

Task :- 25/07/2025

Task 1 :- **1 S3 Bucket Replication (CRR)**

- ❖ Create a second bucket in another region
- ❖ Enable Cross-Region Replication to replicate files automatically



1) Open the AWS console

Then

sign in

Then

Search in bar **S3**

Then

Click on Create bucket button

The screenshot shows the AWS S3 console interface. On the left, there's a navigation sidebar with options like EC2, IAM, S3, CloudWatch, and Amazon S3. The main area is titled 'General purpose buckets' and shows two existing buckets: 'sai-prasad-paper-products.in' and 'samartha-resume-bucket'. A prominent orange 'Create bucket' button is located at the top right of this section. Below the buckets, there are two cards: 'Account snapshot' (updated daily) and 'External access summary - new' (updated daily). At the bottom of the page, there are links for CloudShell, Feedback, and various AWS terms like Privacy, Terms, and Cookie preferences.

Then

In **General configuration** at
AWS region :- **Asia Pacific (Mumbai) ap-south-1**

In **Bucket type**

Select the **General Purpose** option

The screenshot shows the 'Create bucket' page in the AWS Management Console. The 'General configuration' section is visible. Under 'Bucket type', the 'General purpose' option is selected, indicated by a blue border and a checked radio button. A red arrow points to this selection. The 'Directory' option is also present with its description. Other fields like 'Bucket name' (set to 'myawsbucket') and 'Copy settings from existing bucket - optional' are shown. The bottom of the screen includes standard AWS navigation and footer links.

Then

In **Bucket name** :- Type name of Bucket

Name = **yash-first**

This screenshot is identical to the one above, showing the 'Create bucket' page. The 'General configuration' section is displayed, focusing on the 'Bucket name' field which contains 'yash-first'. A red arrow points to this field. The rest of the page, including the 'General purpose' bucket type selection, is the same as the previous screenshot.

Then

In **Object Ownership** select the option **ACLs enabled**

The screenshot shows the 'Object Ownership' section of the 'Create bucket' wizard. It includes two radio button options: 'ACLs disabled (recommended)' and 'ACLs enabled'. The 'ACLs enabled' option is selected and highlighted with a blue border. A red arrow points to this selection. Below the radio buttons is a warning message: '⚠️ We recommend disabling ACLs, unless you need to control access for each object individually or to have the object writer own the data they upload. Using a bucket policy instead of ACLs to share data with users outside of your account simplifies permissions management and auditing.' At the bottom of the section, there's a note: 'If you want to enforce object ownership for new objects only, your bucket policy must specify that the bucket-owner-full-control canned ACL is required for object uploads. [Learn more](#)'.

Then

In Block Public Access setting for this bucket untick the option
Block all public access

The screenshot shows the 'Block Public Access settings for this bucket' section. It contains a list of four checkboxes under the heading 'Block all public access'. The first checkbox, 'Block all public access', is checked and has a red arrow pointing to it. The other three checkboxes are unchecked. The descriptions for the unchecked options are: 'Block public access to buckets and objects granted through new access control lists (ACLS)', 'Block public access to buckets and objects granted through any access control lists (ACLS)', 'Block public access to buckets and objects granted through new public bucket or access point policies', and 'Block public and cross-account access to buckets and objects through any public bucket or access point policies'.

Then

When we untick the option there below one message come , ticket that block.

The screenshot shows the AWS S3 'Create bucket' settings page. In the 'Block Public Access settings for this bucket' section, there are several options under 'Block all public access'. One option is checked: 'I acknowledge that the current settings might result in this bucket and the objects within becoming public.' A red arrow points to this checkbox.

Then
In Bucket Versioning select the option Enable

The screenshot shows the AWS S3 'Create bucket' settings page. In the 'Bucket Versioning' section, the 'Enable' radio button is selected, indicated by a red arrow. A red box highlights the 'I acknowledge that the current settings might result in this bucket and the objects within becoming public.' checkbox, which is also checked.

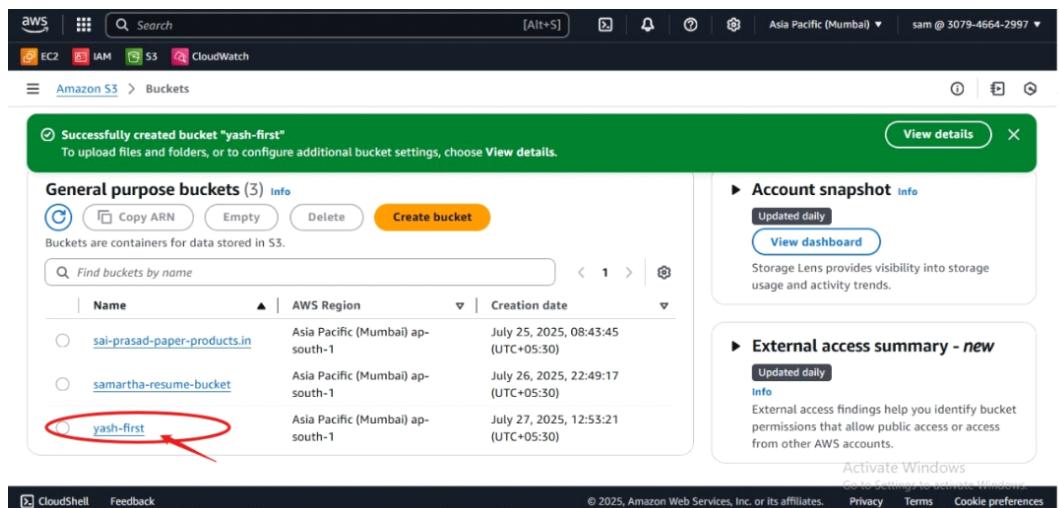
Then
Click on Create Bucket button

The screenshot shows the AWS S3 'Create bucket' settings page with the following configurations:

- Encryption: SSE-KMS (radio button selected)
- Bucket Key: Enabled (radio button selected)
- Advanced settings: Enabled (checkbox selected)
- Note: 'After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.'

A red arrow points to the 'Create bucket' button at the bottom right of the page.

Then
See in
Bucket created successfully
yash-first



The screenshot shows the AWS S3 Buckets page. At the top, there is a green banner with the message "Successfully created bucket 'yash-first'". Below the banner, the "General purpose buckets" section lists three buckets:

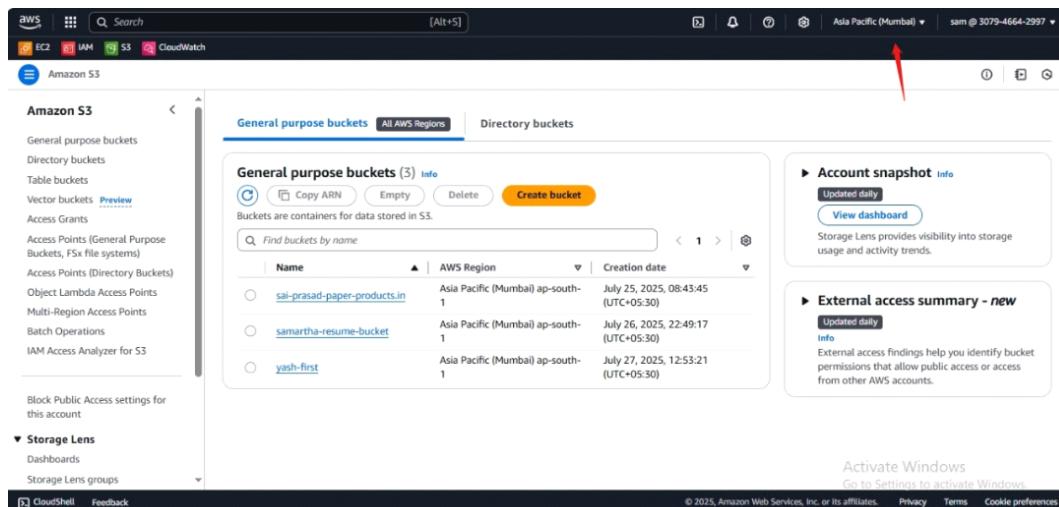
| Name | AWS Region | Creation date |
|------------------------------|----------------------------------|-------------------------------------|
| sai-prasad-paper-products.in | Asia Pacific (Mumbai) ap-south-1 | July 25, 2025, 08:43:45 (UTC+05:30) |
| samartha-resume-bucket | Asia Pacific (Mumbai) ap-south-1 | July 26, 2025, 22:49:17 (UTC+05:30) |
| yash-first | Asia Pacific (Mumbai) ap-south-1 | July 27, 2025, 12:53:21 (UTC+05:30) |

A red arrow points to the "yash-first" bucket entry.

Now we will create a **new bucket in new region**.

So

First Click on **Region** and select **new region**
Old region :- **Asia Pacific (Mumbai) ap-south-1**



The screenshot shows the AWS S3 Buckets page. A red arrow points to the "All AWS Regions" dropdown menu in the top navigation bar. The "General purpose buckets" section lists the same three buckets as before:

| Name | AWS Region | Creation date |
|------------------------------|----------------------------------|-------------------------------------|
| sai-prasad-paper-products.in | Asia Pacific (Mumbai) ap-south-1 | July 25, 2025, 08:43:45 (UTC+05:30) |
| samartha-resume-bucket | Asia Pacific (Mumbai) ap-south-1 | July 26, 2025, 22:49:17 (UTC+05:30) |
| yash-first | Asia Pacific (Mumbai) ap-south-1 | July 27, 2025, 12:53:21 (UTC+05:30) |

Selected new region

Changed to new region :- Asia Pacific (Singapore) ap-southeast-

1

The screenshot shows the AWS S3 console interface. The top navigation bar includes links for EC2, IAM, S3, CloudWatch, and Amazon S3. The region dropdown at the top right is set to "Asia Pacific (Singapore)". The main content area displays "General purpose buckets" with three entries: "sai-prasad-paper-products.in" (Region: Asia Pacific (Mumbai) ap-south-1, Creation date: July 25, 2025, 08:43:45 UTC+05:30), "samartha-resume-bucket" (Region: Asia Pacific (Mumbai) ap-south-1, Creation date: July 26, 2025, 22:49:17 UTC+05:30), and "yash-first" (Region: Asia Pacific (Mumbai) ap-south-1, Creation date: July 27, 2025, 12:53:21 UTC+05:30). To the right of the bucket list are two informational boxes: "Account snapshot" and "External access summary - new". The bottom right corner of the page includes standard AWS footer links: Activate Windows, Go to Settings to activate Windows, Privacy, Terms, and Cookie preferences.

Then

Create bucket click button

This screenshot is identical to the one above, showing the AWS S3 console with the region set to "Asia Pacific (Singapore)". The "Create bucket" button in the top right of the main content area is highlighted with a red arrow. The rest of the interface, including the bucket list and informational boxes, remains the same.

Then

Do same process as done first bucket.

Also do enable to the Bucket Versioning keep it enable.

I forgot to do enable done after so added this image for reference

The screenshot shows the 'Create replication rule' wizard in the AWS S3 console. The 'Destination' step is selected. A red arrow points to the 'Bucket name' input field, which contains 'yash-second'. A callout box highlights the 'Object versioning enabled' section, stating: 'This bucket now has object versioning enabled. If you need to suspend versioning you can do so in [Bucket properties](#) and you will no longer be able to use it as a destination bucket for this rule.' The 'Destination Region' is set to 'Asia Pacific (Singapore) ap-southeast-1'. At the bottom, there are links for CloudShell, Feedback, and a copyright notice.

Then

Second Bucket created successfully

The screenshot shows the AWS S3 buckets list. A green banner at the top says 'Successfully created bucket "yash-second"'. Below it, the 'General purpose buckets' tab is selected, showing four buckets: 'sai-prasad-paper-products.in', 'samartha-resume-bucket', 'yash-first', and 'yash-second'. The 'yash-second' bucket is circled in red. A red arrow points from the text above to this circled bucket. On the right side, there are sections for 'Account snapshot' and 'External access summary'. At the bottom, there are links for CloudShell, Feedback, and a copyright notice.

Then

Click on first bucket (yash-first)

The screenshot shows the AWS S3 Management console for a bucket named 'yash-first'. The 'Management' tab is highlighted with a red circle. The left sidebar contains links for General purpose buckets, Storage Lens, and CloudShell. The main area displays a table with columns for Name, Type, Last modified, Size, and Storage class, showing 'No objects'.

Then go to **Management** option & click it

This screenshot is identical to the one above, showing the AWS S3 Management console for the 'yash-first' bucket. The 'Management' tab is again highlighted with a red circle. The interface and data are consistent with the first screenshot.

Then

Scroll down & go to **Replication rules**

The screenshot shows the AWS S3 Buckets page for a bucket named 'yash-first'. On the left, there's a sidebar with various services like EC2, IAM, S3, and CloudWatch. The main content area has tabs for 'Lifecycle rule na...', 'Status', 'Scope', 'Current version ...', 'Noncurrent vers...', 'Expired object d...', and 'Incomplete mul...'. Below this, it says 'No lifecycle rules' and 'There are no lifecycle rules for this bucket.' with a 'Create lifecycle rule' button. The next section is 'Replication rules (0)', which is circled in red with an arrow pointing to it. It contains a table header with columns: Replication rule name, Status, Destination bucket, Destination Region, Priority, Scope, Storage class, Replica owner, and Replication Time Control. Below the table, it says 'No replication rules' and 'You don't have any rules in the replication configuration.' with a 'Create replication rule' button. At the bottom, there's an 'Inventory configurations (0)' section.

Then click on option **Create replication rule**

This screenshot is similar to the previous one but focuses on the 'Create replication rule' button. A red arrow points to the 'Create replication rule' button in the 'Actions' dropdown menu under the 'Replication rules (0)' section. The rest of the interface is identical to the first screenshot.

Then In **Replication rule configuration at Replication rule name**
Write name :- rule1

Create replication rule [Info](#)

Replication rule configuration

Replication rule name
rule1 rule1

Status
 Enabled
 Disabled

Priority
The priority value resolves conflicts that occur when an object is eligible for replication under multiple rules to the same destination. The rule is added to the configuration at the highest priority and the priority can be changed on the replication rules table.
0

Source bucket
Source bucket name
yash-first
Source Region
Asia Pacific (Mumbai) ap-south-1

Activate Windows
Go to Settings to activate Windows.

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Then

In **Source bucket** at **Choose a rule scope** in their
Select option 2nd option **apply to all object in the bucket** click it

Amazon S3 > Buckets > yash-first > Replication rules > Create replication rule

The priority value resolves conflicts that occur when an object is eligible for replication under multiple rules to the same destination. The rule is added to the configuration at the highest priority and the priority can be changed on the replication rules table.
0

Source bucket
Source bucket name
yash-first
Source Region
Asia Pacific (Mumbai) ap-south-1

Choose a rule scope
 Limit the scope of this rule using one or more filters
 Apply to all objects in the bucket

Destination

Activate Windows
Go to Settings to activate Windows.

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Then

In **Destination** at Choose 1st option **Choose a bucket in account**

Then

In **Bucket name**

Search second bucket name :- **yash-second**

The screenshot shows the 'Create replication rule' step in the AWS S3 console. In the 'Destination' section, under 'Bucket name', the value 'yash-second' is entered, highlighted with a red arrow. A note below states: 'Object versioning enabled. This bucket now has object versioning enabled. If you need to suspend versioning you can do so in Bucket properties and you will no longer be able to use it as a destination bucket for this rule.' Under 'Destination Region', 'Asia Pacific (Singapore) ap-southeast-1' is selected. At the bottom, there are links for CloudShell, Feedback, and a copyright notice.

Then

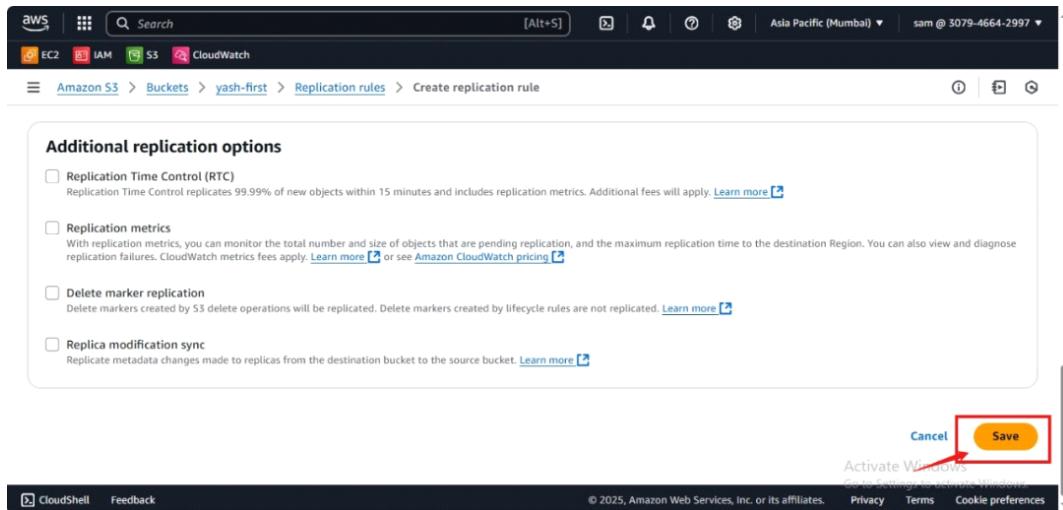
In IAM role at Select 1st option **Create new role**

The screenshot shows the 'Create replication rule' step in the AWS S3 console. In the 'IAM role' section, the radio button for 'Create new role' is selected, highlighted with a red arrow. Below it are two other options: 'Choose from existing IAM roles' and 'Enter IAM role ARN'. Under 'Encryption', the checkbox for 'Replicate objects encrypted with AWS Key Management Service (AWS KMS)' is checked. At the bottom, there are links for CloudShell, Feedback, and a copyright notice.

Then keep all as it is

Then

Click on **Save** button



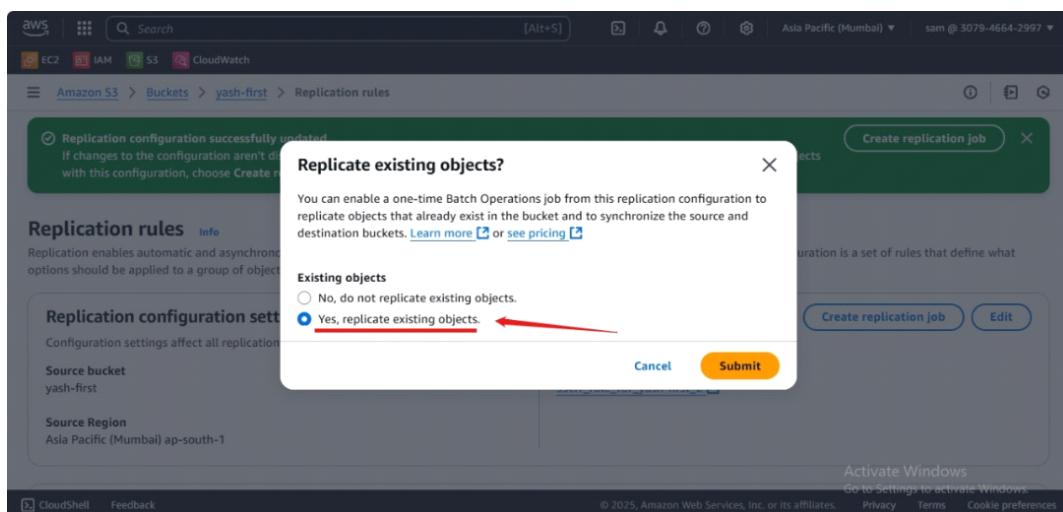
Then

Replicate existing object new screen will open Choose their the option 2nd **Yes, replicate existing objects**

This will do works as if in bucket their already exit old file then that file will also save in another bucket.

&

If we keep the 1st option as it is then the old file will transfer or show in new bucket which we will replicate, only new file will transfer & show.



Then

Click on submit button

Then

Go to **yash-first**

Then go to option **Upload** and now upload the files of the website

The screenshot shows the AWS S3 console interface. At the top, there's a navigation bar with links for EC2, IAM, S3, and CloudWatch. Below that, the path 'Amazon S3 > Buckets > yash-first' is shown. The main area is titled 'yash-first info'. Under the 'Objects' tab, it says '(0)' and has a 'Copy S3 URI' button. There are also 'Copy URL', 'Download', 'Open', 'Delete', 'Actions', 'Create folder', and a prominent yellow 'Upload' button with a red arrow pointing to it. Below the toolbar, there's a search bar, a 'Show versions' toggle, and filters for 'Name', 'Type', 'Last modified', 'Size', and 'Storage class'. A message states 'No objects' and 'You don't have any objects in this bucket.' At the bottom, there's a 'CloudShell' link, a feedback link, and copyright information: '© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences'.

Then

The screenshot shows the 'Upload' page for the 'yash-first' bucket. At the top, it says 'Upload info' and 'Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDKs or Amazon S3 REST API. Learn more'. Below that is a large dashed box with the text 'Drag and drop files and folders you want to upload here, or choose Add files or Add folder.' A red arrow points to the 'Add files' button in the top right corner of this box. Below this, there's a table titled 'Files and folders (15 total, 2.6 MB)'. The table lists 15 files and folders with columns for Name, Folder, Type, and Size. A red arrow points to the 'Add folder' button in the top right corner of the table. At the bottom, there's a 'CloudShell' link, a feedback link, and copyright information: '© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences'.

Then

Click on **upload** button

The screenshot shows the AWS CloudWatch Metrics console. A green notification bar at the top says "Upload succeeded" with a link to "Files and folders". Below it, a table titled "Files and folders (15 total, 2.6 MB)" lists 8 files, all of which have a status of "Succeeded".

| Name | Folder | Type | Size | Status | Error |
|---------|-------------|------------|----------|-----------|-------|
| re2.jpg | assets/img/ | image/jpeg | 151.1 KB | Succeeded | - |
| re3.jpg | assets/img/ | image/jpeg | 183.9 KB | Succeeded | - |
| re4.jpg | assets/img/ | image/jpeg | 107.1 KB | Succeeded | - |
| re5.jpg | assets/img/ | image/jpeg | 117.2 KB | Succeeded | - |
| re6.jpg | assets/img/ | image/jpeg | 105.0 KB | Succeeded | - |
| re7.jpg | assets/img/ | image/jpeg | 129.0 KB | Succeeded | - |
| re8.jpg | assets/img/ | image/jpeg | 15.1 KB | Succeeded | - |

Then

In **Object** you can see that the **files are uploaded and save.**

The screenshot shows the AWS S3 Bucket Objects page for the bucket "yash-first". It displays two objects: "assets/" (a folder) and "services.html" (an HTML file). An arrow points to the "Actions" button in the toolbar above the list.

| Name | Type | Last modified | Size | Storage class |
|---------------|--------|-------------------------------------|---------|---------------|
| assets/ | Folder | - | - | - |
| services.html | html | July 27, 2025, 18:06:05 (UTC+05:30) | 14.3 KB | Standard |

Then

Go to **Action** option

Then

Click on **Action** button, their scroll down & click on option
on **Make public using ACL**

Click it

The screenshot shows the AWS S3 console interface. At the top, there's a navigation bar with various service icons like EC2, IAM, S3, and CloudWatch. Below it, the path 'Amazon S3 > Buckets > yash-first' is visible. The main area is titled 'yash-first' with a 'Info' link. A horizontal menu bar below the title includes 'Objects', 'Properties', 'Permissions', 'Metrics', 'Management', and 'Access Points'. The 'Objects' tab is selected. Underneath, a table lists two objects: 'assets/' (Folder) and 'services.html' (html). To the right of the table is a large 'Actions' dropdown menu, which is highlighted with a red box. The menu contains options like 'Move', 'Initiate restore', 'Query with S3 Select', 'Edit actions', 'Rename object', 'Edit storage class', 'Edit server-side encryption', 'Edit metadata', 'Edit tags', and 'Make public using ACL'. Red arrows point from the text 'Click on Make public button' to the 'Make public' button in the dropdown menu.

Then

Click on **Make public** button

The screenshot shows the 'Make public' confirmation dialog in the AWS S3 console. The URL in the browser is 'Amazon S3 > Buckets > yash-first > Make public'. The page title is 'Make public' with an 'Info' link. It says 'The make public action enables public read access in the object access control list (ACL) settings.' Below this is a warning box with a yellow exclamation mark:

- When public read access is enabled and not blocked by Block Public Access settings, anyone in the world can access the specified objects.
- This action applies to all objects within the specified folders. Objects added to these folders while the action is in progress might be affected.

At the bottom, there's a 'Specified objects' section with a table showing the same two objects: 'assets/' and 'services.html'. To the right of the table are 'Cancel' and 'Make public' buttons. The 'Make public' button is highlighted with a red box. Red arrows point from the text 'Click on Make public button' to the 'Make public' button in the dialog.

Then

It will successfully be public now.

Then go to **Properties** option

Click it

The screenshot shows the AWS S3 Bucket Properties page for 'yash-first'. At the top, there's a navigation bar with tabs like 'Objects', 'Properties' (which is selected), 'Permissions', 'Metrics', 'Management', and 'Access Points'. Below this, the 'Bucket overview' section displays basic information: AWS Region (Asia Pacific (Mumbai) ap-south-1), Amazon Resource Name (ARN) (arn:aws:s3:::yash-first), and Creation date (July 27, 2025, 12:53:21 (UTC+05:30)). The 'Bucket Versioning' section is shown with 'Enabled' status. Under 'Multi-factor authentication (MFA) delete', it says 'Disabled'. At the bottom, there are links for 'CloudShell', 'Feedback', and copyright information.

Then scroll down & go to last option **Static website hosting**.
Their the **Edit** button is click it

This screenshot shows the 'Static website hosting' configuration section. It includes a note about AWS Amplify Hosting and a 'Create Amplify app' button. The 'Edit' button for this section is highlighted with a red box and a red arrow pointing to it. Below this, there's a note about S3 static website hosting being disabled.

Then
In **Static website hosting** select **Enable** option.
Then
In **Hosting type** select first option **Host a static website**

Edit static website hosting [Info](#)

Static website hosting

Use this bucket to host a website or redirect requests. [Learn more](#)

Static website hosting

Disable

Enable ←

Hosting type

Host a static website Use the bucket endpoint as the web address. [Learn more](#) ←

Redirect requests for an object Redirect requests to another bucket or domain. [Learn more](#)

ⓘ For your customers to access content at the website endpoint, you must make all your content publicly readable. To do so, you can edit the S3 Block Public Access settings for the bucket.

For more information, see [Using Amazon S3 Block Public Access](#)

Index document

Specify the home or default page of the website.

Activate Windows
Go to Settings to activate Windows

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Then

In **Index document** add the file name which is main file .

Add the file that which is your website main file

Main File name :- **services.html**

Then

Click on **Save changes** button

yash-first [Info](#)

Properties [Objects](#) [Permissions](#) [Metrics](#) [Management](#) [Access Points](#)

Bucket overview

AWS Region: Asia Pacific (Mumbai) ap-south-1

Amazon Resource Name (ARN): arn:aws:s3:::yash-first

Creation date: July 27, 2025, 12:53:21 (UTC+05:30)

Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

Bucket Versioning: Enabled

Edit

ⓘ Multi-factor authentication (MFA) delete

Activate Windows
Go to Settings to activate Windows

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Then go to **properties** option & scroll down till **static website hosting** we will get the **link of website**, click on it then new browser will open and our **website will display** .

Output of website :-

Screenshot of page :-

The screenshot shows a mobile-optimized website for 'Machinery Maintenance & Repairing Services'. At the top, there is a circular logo with a yellow and red design, followed by the text 'OUR SERVICES' in white on an orange gradient button and 'HOME' in small blue text. Below this, the main title 'Machinery Maintenance & Repairing Services' is centered. The page features three yellow callout boxes: 'Regular Maintenance' (with a wrench icon), 'Repair Services' (with a gear icon), and 'Spare Parts Supply' (with a building icon). Each box contains a brief description. At the bottom, there is a horizontal image of a workshop or factory floor.

Machinery Maintenance & Repairing Services

Regular Maintenance
We offer periodic maintenance to keep your machines running smoothly.

Repair Services
Our expert team provides quick and efficient repair solutions for all types of paper machinery.

Spare Parts Supply
We supply high-quality spare parts to ensure the durability of your machines.

Full page screenshot :-



Machinery Maintenance & Repairing Services



Regular Maintenance

We offer periodic maintenance to keep your machines running smoothly.



Repair Services

Our expert team provides quick and efficient repair solutions for all types of paper machinery.



Spare Parts Supply

We supply high-quality spare parts to ensure the durability of your machines.



• • • • •

Schedule a Maintenance Check



Full Name

SAMARTHA SANTOSH LASURE

Email

samarth.lasure@gmail.com

Phone

123456789

Describe the Issue

hi

Submit Request

Then

Go back and click On second bucket:- **yash-second**

Their we can see that in **object the files** are already there.

Then

Go to **Properties** & scroll down at bottom till **static website hosting** we can see that the **website link** is showing.

The screenshot shows the AWS S3 Bucket Properties page for a bucket named 'yash-second'. In the 'Static website hosting' section, there is a callout box with the text: 'We recommend using AWS Amplify Hosting for static website hosting. Deploy a fast, secure, and reliable website quickly with AWS Amplify Hosting. Learn more about Amplify Hosting or View your existing Amplify apps.' Below this, it says 'S3 static website hosting' is Enabled. A red arrow points to the 'Bucket website endpoint' field, which contains the URL 'http://yash-second.s3-website-ap-southeast-1.amazonaws.com'. The left sidebar shows navigation options like General purpose buckets, Storage Lens, and IAM Access Analyzer.

It means we have successfully done CRR.

Task 2 :- S3 Cross-Account Access

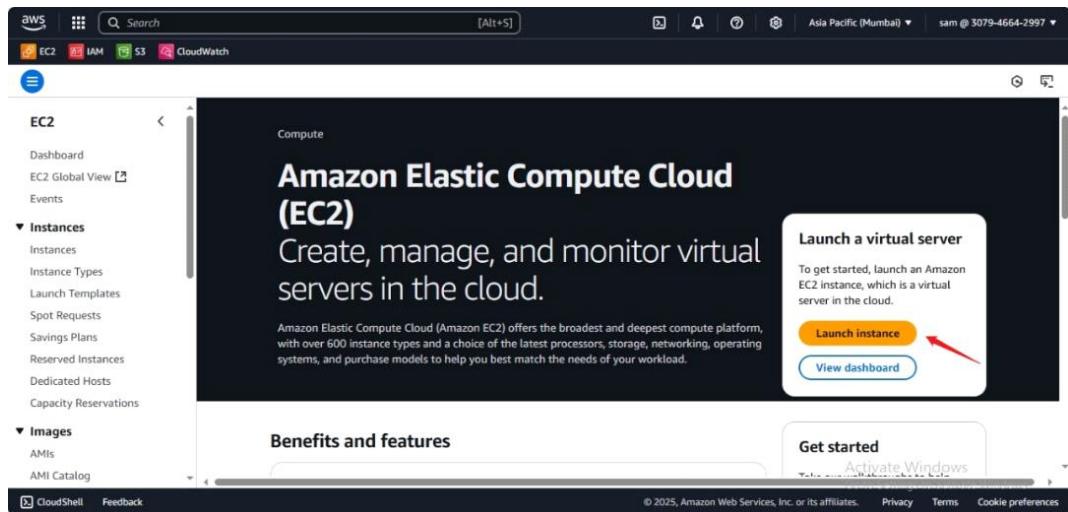
- ◆ Create a bucket in one AWS account
- ◆ Grant access to another account using a bucket policy

Task 2 :- Can't do because new / another account is not opening problem perist.

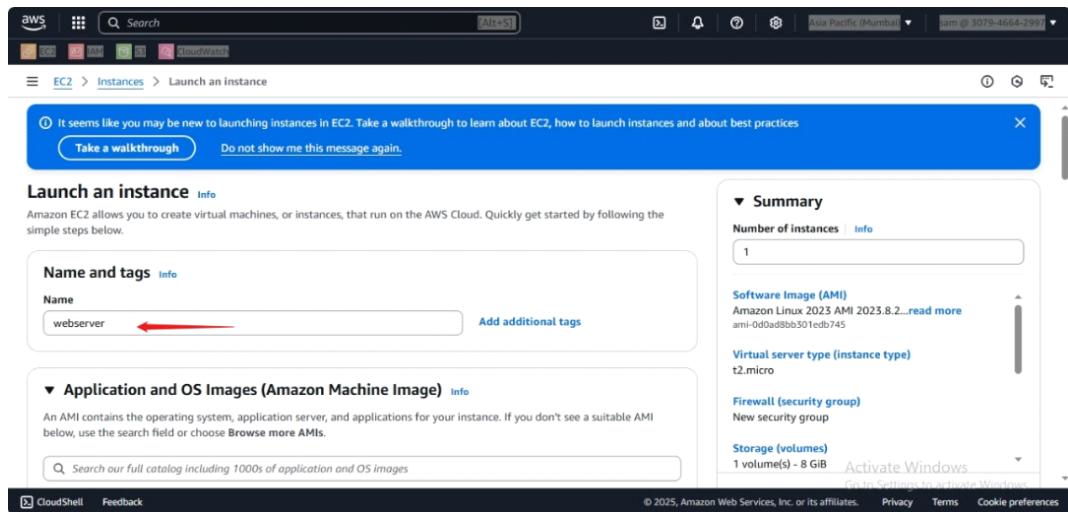
Task 3:- ◆ Auto scaling with Load Balancer

Go to AWS console

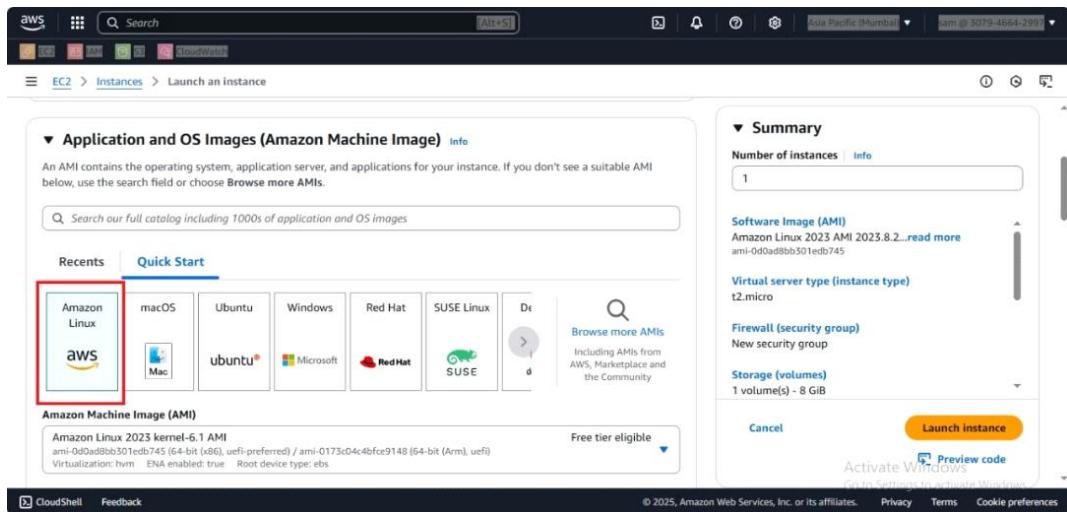
Then
Search Ec2
Then click it
Then
Click on **Launch Instances**



Then
Instances name :- webserver



Then
In Application &OS Images (AMI) in this Select
Amazon Linux



Then

In **Instance type** keep as it is

because it defaulttly select the free tier

So **t2.micro**

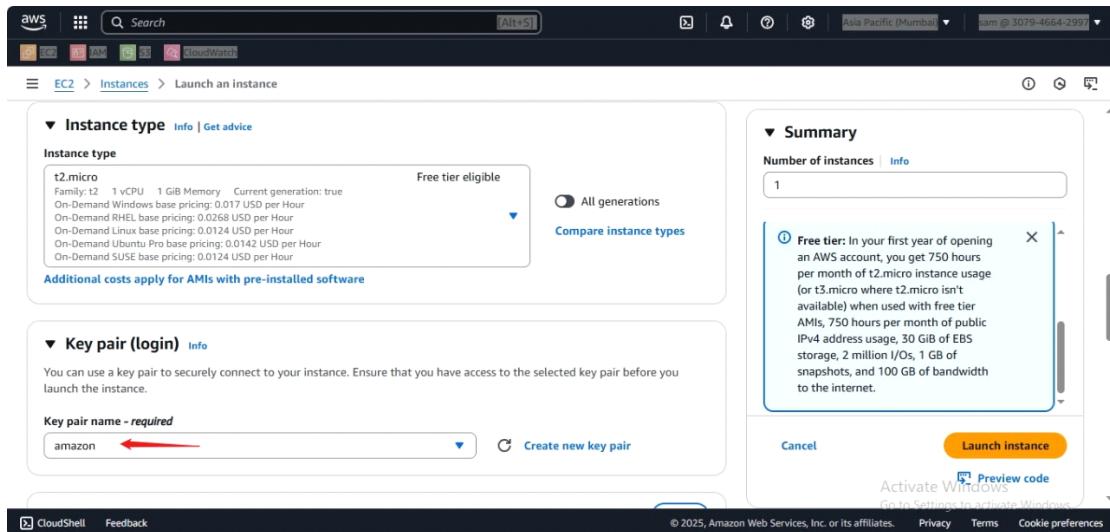
Then

In **Key pair**

Keypair name in drop-down select the key or make new key

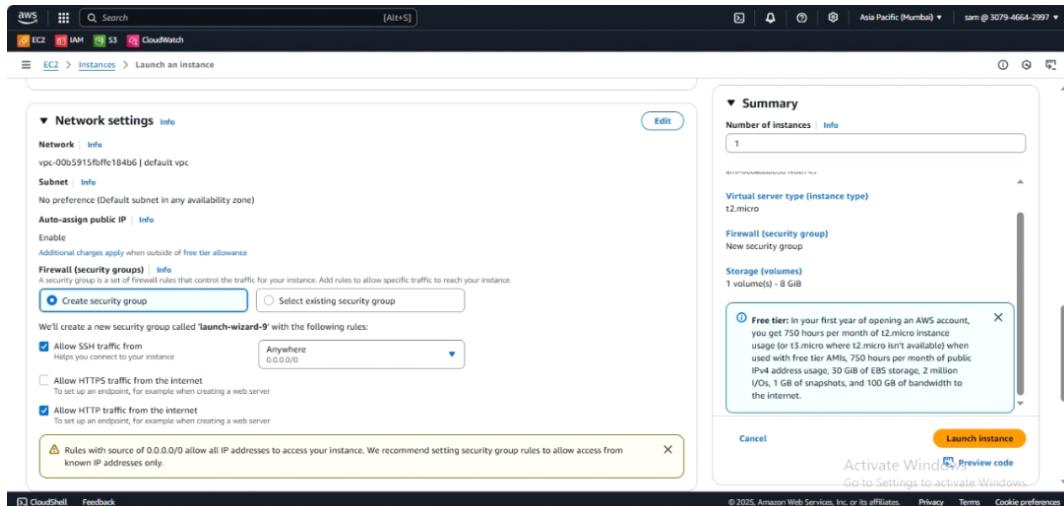
I will select the old one key, i have already.

Name :- amazon.pem



Then

In **Network setting** at firewall (security group) select the option of **Create security group** & allow/tick the option of **SSH & HTTP**



Then

Added the code / script

```
#!/bin/bash
```

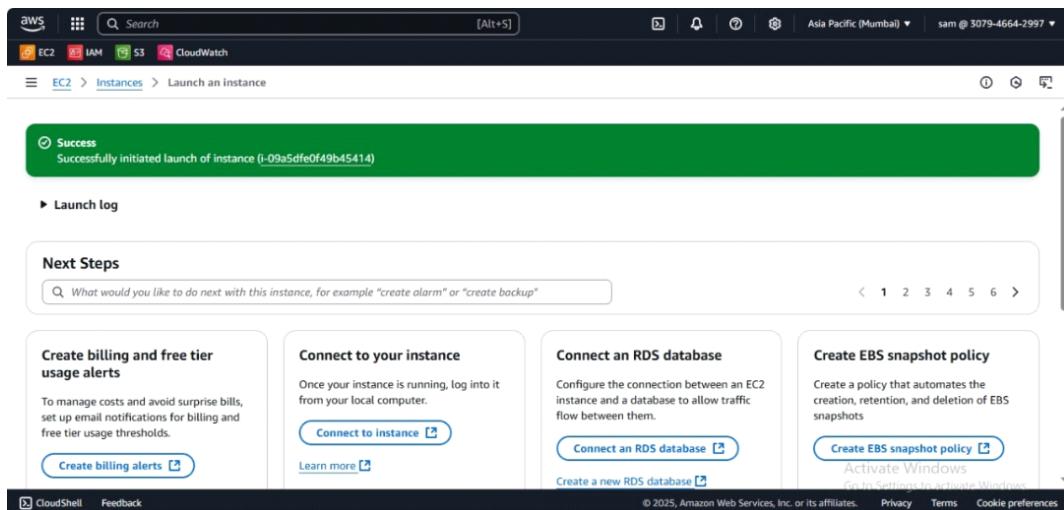
```
yum update -y
```

```
yum install -y httpd
```

```
echo " hellow from auto scaling "> /var/www/html/index.html
```

```
systemctl start httpd  
systemctl enable httpd
```

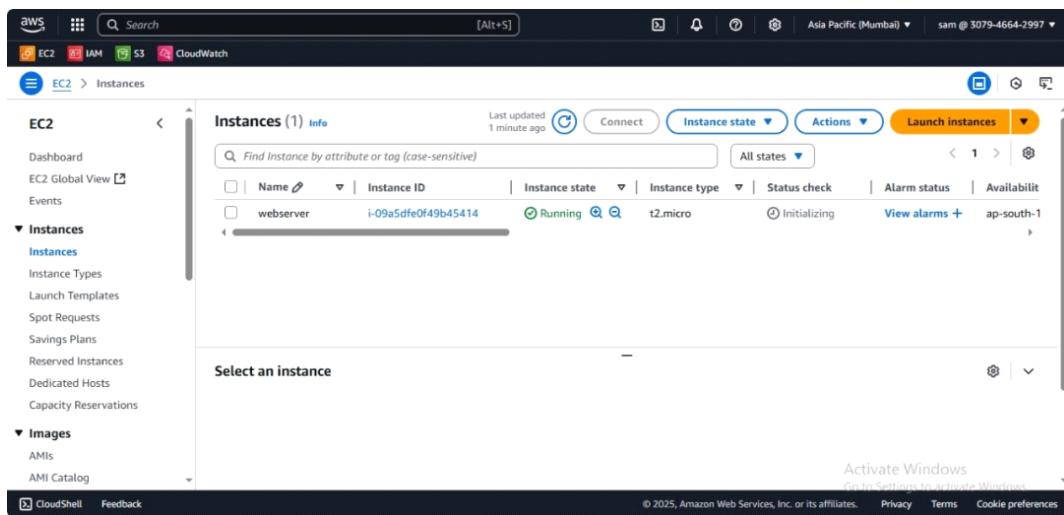
Then click on **Launch Instances** button



Then

Wait for **instance state** to be in **running** state

Then



Then
Click on instance id

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with options like Dashboard, EC2 Global View, Events, Instances (selected), Images, and CloudShell. The main area shows a table with one row for an instance named 'webserver' with the ID 'i-09a5dfe0f49b45414'. The instance is listed as 'Running' with type 't2.micro'. A red arrow points to the 'Instance ID' column.

Then
Copy the Public IPv4 address

The screenshot shows the 'Instance summary for i-09a5dfe0f49b45414 (webserver)' page. The left sidebar is identical to the previous screenshot. The main content area displays various details about the instance, including its Public IPv4 address (65.1.107.66) and Private IP address (172.31.11.186). A red arrow highlights the Public IPv4 address field.

Then copy the Public IPv4 address paste it to browser & see the File output

Hello from Auto Scaling!



Activate Windows
Go to Settings to activate Windows.

Then we can see the output of the file.

Now we want to create an AMI (Amazon Machine Image) :-

Go to **EC2** & select the created **instances** & click on it (**webserver**)

Then

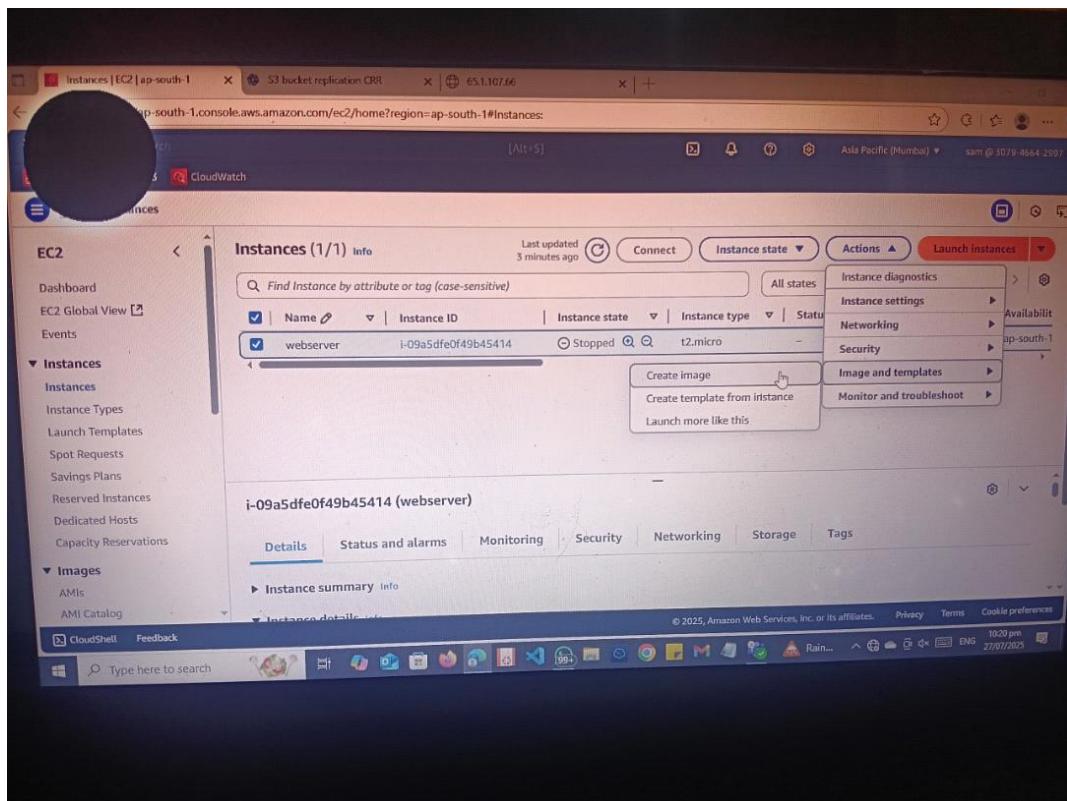
Go to **Action** button click it

Then

In this go to option **Image & template** click it

Then

In this **Create image** option click



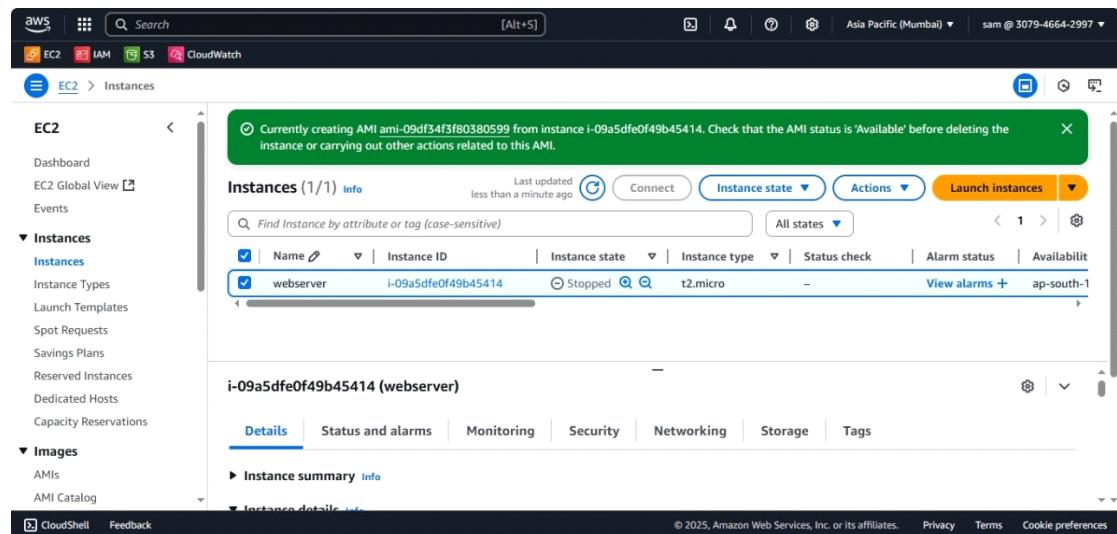
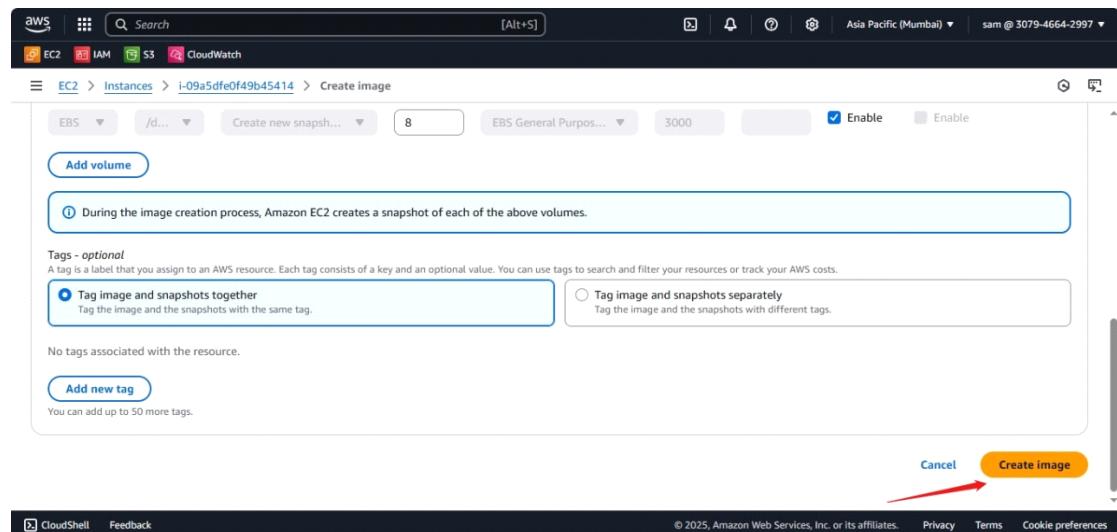
Then

In Create image at **Image details , Instance ID** below is **Image name** at there name it
Name :- webserverAMI

A screenshot of the 'Create image' wizard. The top navigation bar shows 'aws' and 'Search' along with other AWS services like EC2, IAM, S3, and CloudWatch. The current path is 'EC2 > Instances > i-09a5dfe0f49b45414 > Create image'. The main form is titled 'Create image' with 'Info' sub-titled. It says 'An image (also referred to as an AMI) defines the programs and settings that are applied when you launch an EC2 instance. You can create an image from the configuration of an existing instance.' Under 'Image details', there's a 'Instance ID' field with 'i-09a5dfe0f49b45414 (webserver)' and an 'Image name' field with 'webserverAMI' (which has a red arrow pointing to it). Below these are 'Image description - optional' and 'Reboot instance' checkboxes. At the bottom, there's a section for 'Instance volumes' and a footer with 'CloudShell Feedback' and copyright information.

Then

Scroll down till bottom & click on button of **Create image**



Then

our AMI Image is created successfully

&

Wait till the status of AMI image is Available

Amazon Machine Images (AMIs) (1/1) Info

| Name | AMI name | AMI ID | Source | Owner | Visibility | Status | Creation | |
|--|--------------|-----------------------|---------------------------|--------------|------------|--------|-----------|---------|
| <input checked="" type="checkbox"/> webserverAMI | webserverAMI | ami-09df34f3f80380599 | 307946642997/webserverAMI | 307946642997 | Private | | Available | 2025/07 |

Then

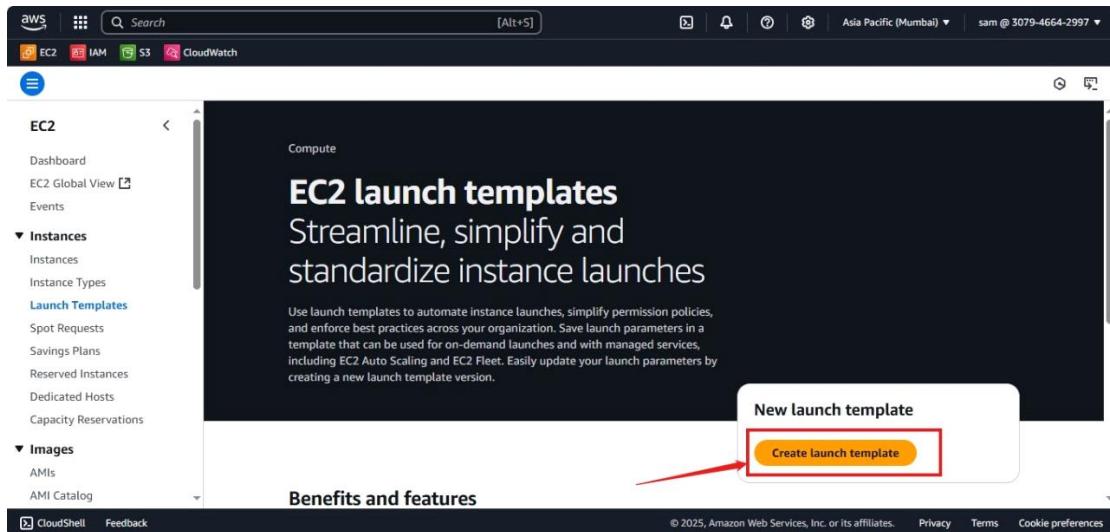
Go to **EC2** in their at left side see **Launch Templates** is there click it

Instances (1) Info

| Name | Instance ID | Instance state | Instance type | Status check | Alarm status | Availability |
|------------------------------------|---------------------|----------------|---------------|--------------|--------------|--------------|
| <input type="checkbox"/> webserver | i-09a5dfe0f49b45414 | | t2.micro | | | ap-south-1 |

Then

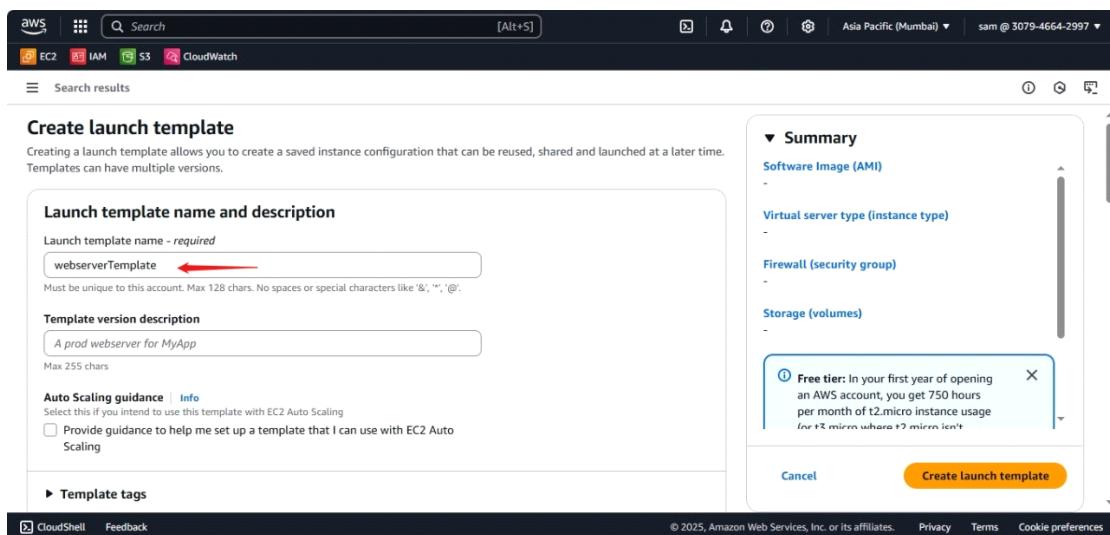
Click on **Create Launch Templates** button



Then

In **Create launch template** at **Launch template name & description**

Name :- webserverTemplate



Then

In **Application & OS Images AMI (Amazon Machine Image)**

Add there our created AMI image

Name :- webserverAMI

click on option **My AMIs**

Then select our **AMI image** created

Application and OS Images (Amazon Machine Image) Info

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose [Browse more AMIs](#).

Search our full catalog including 1000s of application and OS images

Recent AMIs: [Don't include in launch template](#) [Owned by me](#) [Shared with me](#)

Browse more AMIs Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

webserverAMI
ami-09df34f3f80380599
2025-07-27T16:54:28.000Z Virtualization: hvm ENA enabled: true Root device type: ebs Boot mode: uefi-preferred

Description

Architecture [AMI ID](#)

[CloudShell](#) [Feedback](#)

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Then
In **Instance type** select the **t2.micro**

Instance type Info | Get advice

x86_64 ami-09df34f3f80380599

Instance type

t2.micro Family: t2 1 vCPU 1 GiB Memory Current generation: true
Free tier eligible

All generations [Compare instance types](#)

Key pair (login) Info

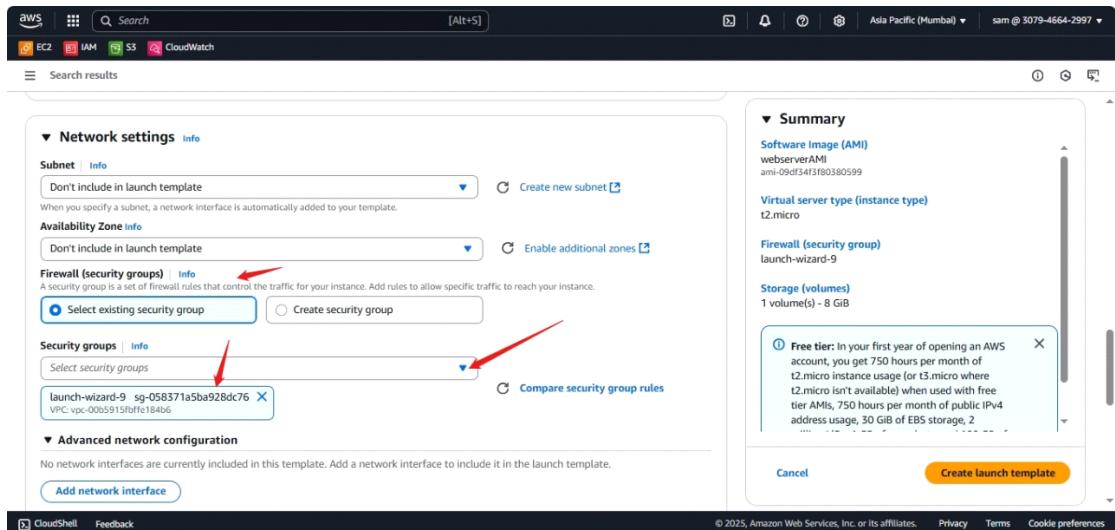
You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name

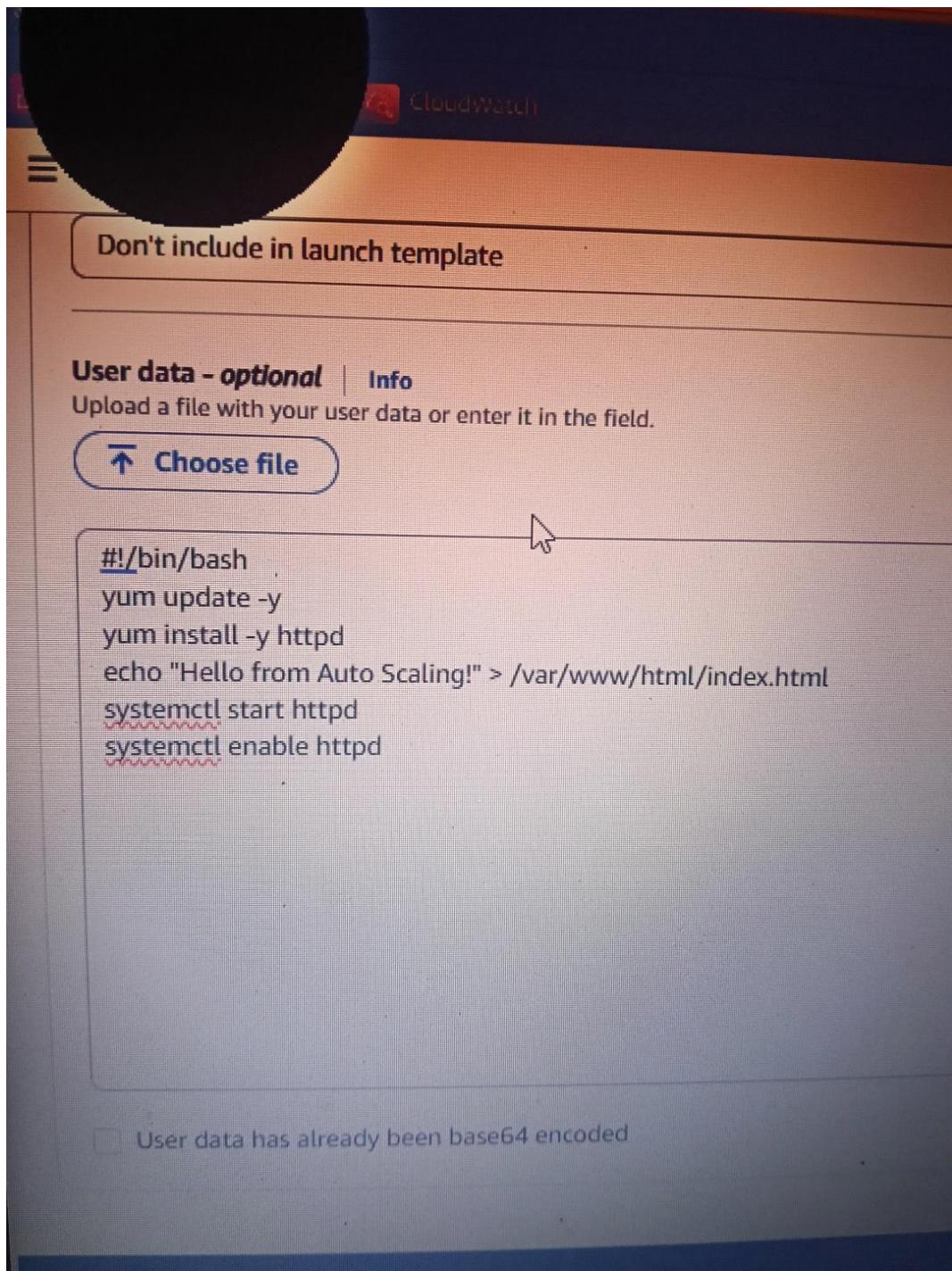
[CloudShell](#) [Feedback](#)

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Then
In **key pair** keep as it is default
Then
In **Network setting at Subnet** keep as it is default , in **Availability zone** keep as it is default, in **Firewall (Security group)** select the **select existing security group**
Then
Click the arrow then select our **security group**
which we created on Instances time



Then
As we paste one script paste same here also



Then

Click on button **Created launch template**

The screenshot shows the AWS EC2 Launch Template creation process. At the top, there's a green success message: "Success Successfully created webserverTemplate(lt-0fc70c4795013ffed).". Below it, there's an "Actions log" section. Under "Next Steps", there are several options: "Launch an instance", "Create an Auto Scaling group from your template", "Create a Spot Fleet", and "Create Auto Scaling group". The "Create Auto Scaling group" option is highlighted with a red arrow. At the bottom, there are links for "CloudShell" and "Feedback", and a copyright notice: "© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences".

Then
launch template created successfully

Now we will make the target group :-

Go to **EC2** on left side scroll little bit down then in **Load balancing** see in it **Target Groups**
Click it

The screenshot shows the AWS EC2 Target groups page. On the left sidebar, under "Load Balancing", the "Target Groups" option is selected and highlighted with a red box. A red arrow points to this selection. On the main page, there's a table titled "Target groups info" with columns: Name, ARN, Port, Protocol, Target type, Load balancer, and VPC ID. A "Create target group" button is located at the top right of the table area. The message "No target groups" is displayed below the table. At the bottom left, it says "0 target groups selected". A red arrow points to the "Select a target group above." instruction.

Then
In **Specify group details at Choose a target type** in this

Choose Instances option

The screenshot shows the 'Specify group details' step of creating a target group. It includes a sidebar with 'Step 1: Specify group details' (selected) and 'Step 2: Register targets'. The main area is titled 'Basic configuration' with a note: 'Settings in this section can't be changed after the target group is created.' It shows three options: 'Instances' (selected), 'IP addresses', and 'Lambda function'. The 'Instances' section contains a bulleted list: 'Supports load balancing to instances within a specific VPC.', 'Facilitates the use of Amazon EC2 Auto Scaling to manage and scale your EC2 capacity.' A red arrow points to the 'Instances' radio button.

Then

In Target group name

Name :- webtargetgroup

Then

Click on Next button

Then in Register targets scroll down & click on create target group button

The screenshot shows the 'Targets' tab for the 'webtargetgroup'. It displays the following details:

| Total targets | Healthy | Unhealthy | Unused | Initial | Draining |
|---------------|---------|-----------|--------|---------|----------|
| 0 | 0 | 0 | 0 | 0 | 0 |

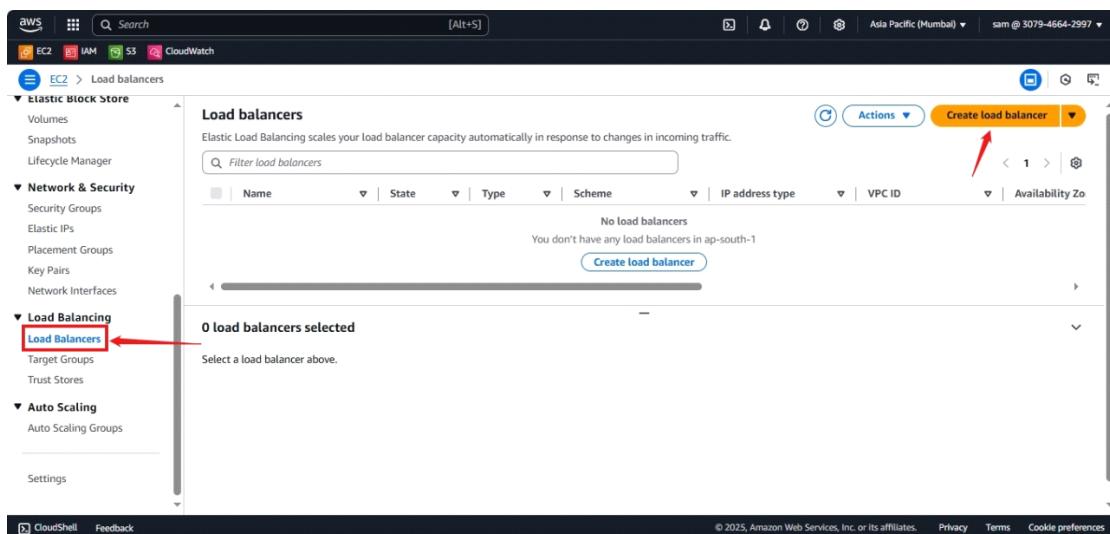
Below the table, there are tabs for Targets, Monitoring, Health checks, Attributes, and Tags. At the bottom, there is a 'Registered targets (0)' section with a 'Register targets' button. A note at the bottom states: 'Targets are automatically registered when they receive the protocol and port specified. Unhealthy targets are not passed on to any registered targets according to the health check settings.'

Then

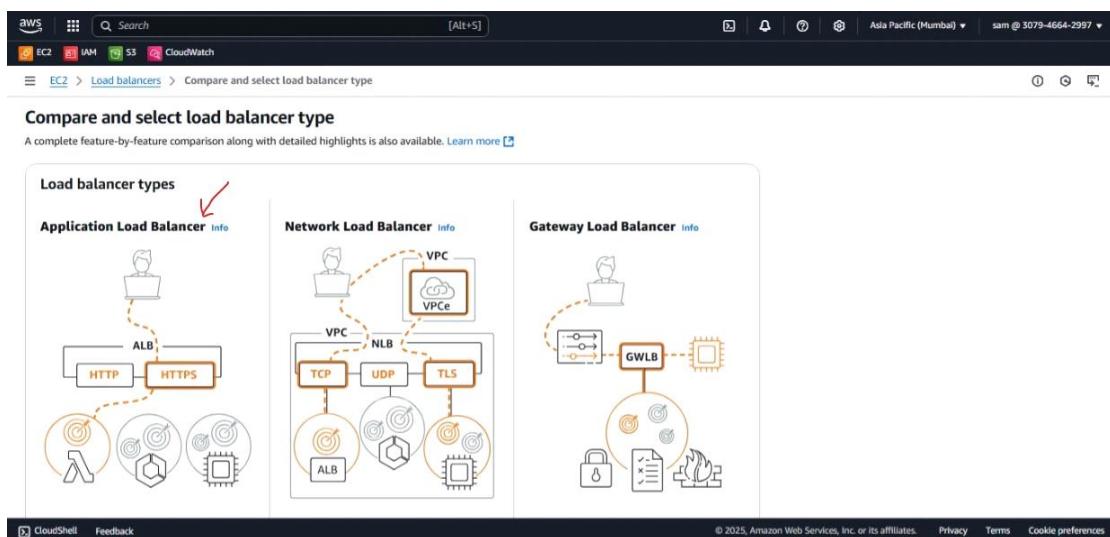
Target group is successful made & ready .

Now we will create a Application Load Balancer :-

Go to **Load balancer** under Load balancing
Then click on **Create load balancer**



Then
In **Compare & select load balancer type**
Select the **Application Load Balancer** & click on **create button** .



Then
Page will open **Create application load balancer**

Then

In Basic Configuration at Load balancer name

Name :- webloadbalancer

Then below it option **Scheme** select in their **Internet-facing** select

Then

In Load balancer IP address type select **IPv4**

The screenshot shows the 'Basic configuration' step of the AWS Create Application Load Balancer wizard. The 'Load balancer name' field is set to 'webloadbalancer'. In the 'Scheme' section, 'Internet-facing' is selected. Under 'Load balancer IP address type', 'IPv4' is selected. The interface includes standard AWS navigation and status bars.

Then

In Network mapping at VPC select **default vpc** & in Availability zone select two zone like ap-south-1a & ap-south-1b

Then

In Listeners & routing

Protocol:- HTTP

Port :- 80

Default action:- select our created target group (webtargetgroup)

The screenshot shows the 'Listeners and routing' step of the AWS Create Application Load Balancer wizard. It displays a single listener configuration for port 80, which is forwarded to the 'webtargetgroup' target group. The interface includes standard AWS navigation and status bars.

Then
Click on **Create load balancer** button

The screenshot shows the AWS EC2 Load Balancers console. A green success message at the top states: "Successfully created load balancer: webloadbalancer. It might take a few minutes for your load balancer to fully set up and route traffic. Targets will also take a few minutes to complete the registration process and pass initial health checks." Below this, a blue info message says: "Application Load Balancers now support public IPv4 IP Address Management (IPAM). You can get started with this feature by configuring IP pools in the Network mapping section." The main panel displays the details of the newly created load balancer "webloadbalancer". The status is "Provisioning". The VPC is "vpc-00b5915fbffe184b6". The hosted zone is "ZP97RAFLXTNZK". Availability zones include "subnet-0b22791683e2e7db6 ap-south-1a (ap-s1-az1)" and "subnet-0ed045f4e1aeeb52 ap-south-1b (ap-s1-az3)". The load balancer IP address type is "IPv4". The DNS name is "webloadbalancer-1069926602.ap-south-1.elb.amazonaws.com (A Record)". The date created was July 27, 2025, 23:31 (UTC+05:30).

Then
Load balancer created successfully

Now we will create an Auto Scaling Group :-

Go to **EC2** on left side scroll down till bottom there will be **Auto Scaling** below it is **Auto Scaling Group** click it .

The screenshot shows the AWS EC2 Auto Scaling groups console. On the left sidebar, under the "Auto Scaling" section, the "Auto Scaling Groups" item is highlighted with a red arrow. The main content area features a large banner with the text "Amazon EC2 Auto Scaling helps maintain the availability of your applications". Below the banner, a call-to-action button says "Create Auto Scaling group" with a red arrow pointing to it. To the right of the button, there's a "How it works" diagram showing an "Auto Scaling group" icon connected to four smaller square icons representing EC2 instances. Other sections visible include "Pricing" and "Getting started".

Then

Click on button **Create Auto Scaling Group**

Then

In **Auto Scaling Group name**

Name :- webASG

Then

In **Launch template** click arrow then select our created webserverTemplate

Then click on **Next** button.

The screenshot shows the 'Choose launch template' step of the 'Create Auto Scaling group' wizard. On the left, a sidebar lists steps from 1 to 7. Step 1 is selected. The main area shows a 'Name' field containing 'webASG' and a note: 'Must be unique to this account in the current Region and no more than 255 characters.' Below it is a 'Launch template' section with a note: 'For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023.'

Then

In **Choose Instance launch option at Network** in their **VPC** select the default vpc then below it **Availability zone** select our selected zone of time load balancer i.e. **ap-south-1a & ap-south-1b**

Then

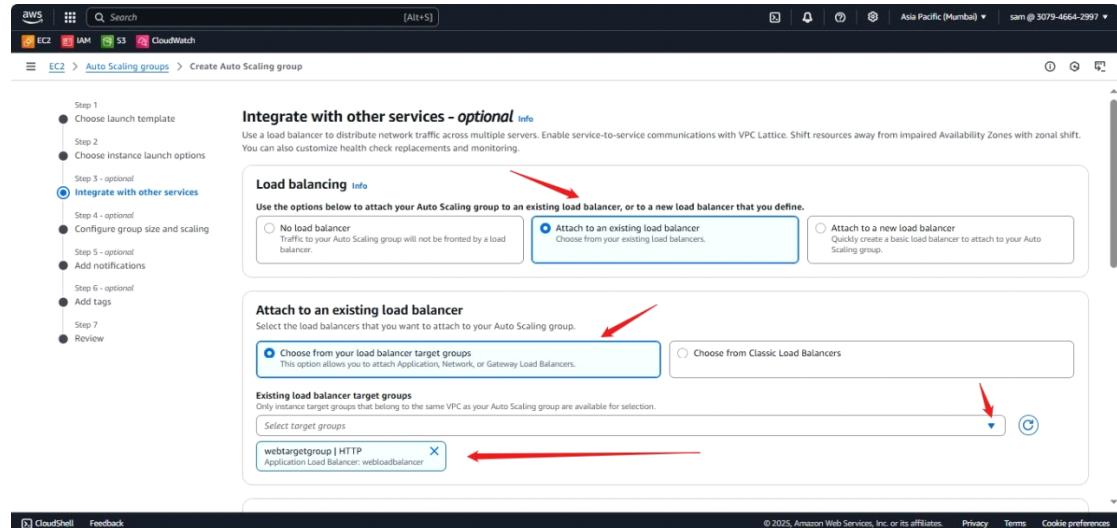
Click on **Next** button

The screenshot shows the 'VPC' configuration step. It starts with a 'VPC' section where 'vpc-00b5915fffe184b6 (default vpc)' is selected. Below it is an 'Availability Zones and subnets' section where two subnets are chosen: 'aps1-az1 (ap-south-1a) | subnet-0b22791683e2e7db6' and 'aps1-az3 (ap-south-1b) | subnet-0ed045f4eb1aeeb52'. At the bottom, there's an 'Availability Zone distribution - new' section with two options: 'Balanced best effort' (selected) and 'Balanced only'. The 'Balanced best effort' option includes a note: 'If launches fail in one Availability Zone, Auto Scaling will attempt to launch in another healthy Availability Zone.' The 'Balanced only' option includes a note: 'If launches fail in one Availability Zone, Auto Scaling will continue to attempt to launch in the unhealthy Availability Zone to preserve balanced distribution.'

Then

In **Integrate with other services** at load balancing select the option 2nd **Attach to an existing load balancer**

Then In Attach to an existing load balancer in their at Existing load balancer target groups click on arrow & select our created target group (**webtargetgroup**).



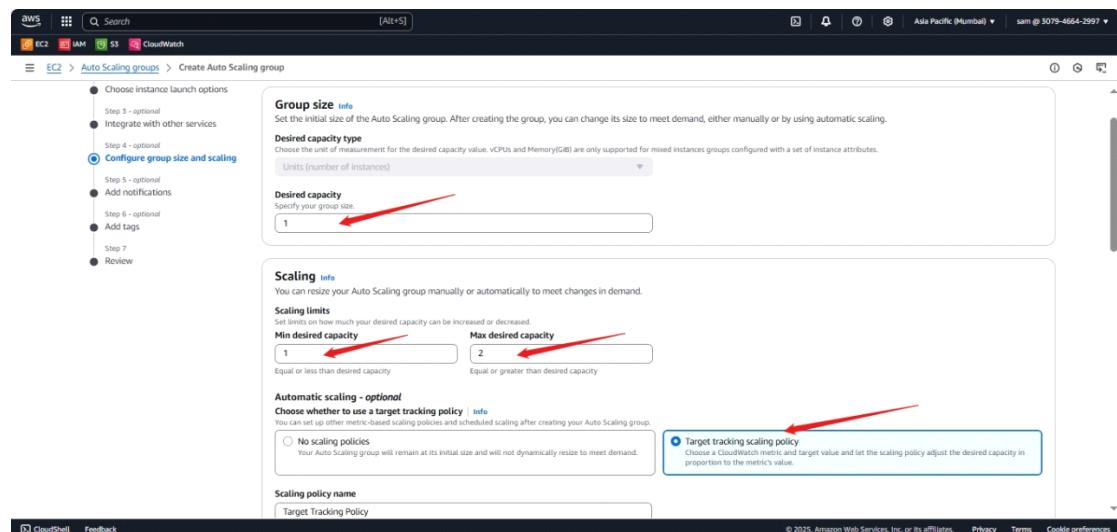
Then click on **Next** button

In **Configure Group size & scaling** at **Group size** below

Desired capacity:- 1 , Min desired capacity:- 1 , Max desired capacity:- 2

Then

In **Automatic scaling** select the option **Target tracking scaling policy**



Then

In **Add notification** skip it & Click on **Next** button

Then

In **Tags** skip it & Click on **Next** button then in **Review** click on **button**

Create Auto Scaling Group

you can see that the group created.

The screenshot shows the AWS Auto Scaling Groups page. At the top, there's a search bar and navigation links for EC2, IAM, S3, and CloudWatch. Below that, the main heading is 'Auto Scaling groups (1) Info'. A search bar says 'Search your Auto Scaling groups'. The table below lists one group: 'webASG' with 'webserverTemplate | Version Default', '0 Instances', 'Status: Updating capacity...', 'Desired capacity: 1', 'Min: 1', 'Max: 2', 'Availability Zones: 2', and 'Creation time: Mon Jul 28 2025 00:02:33 GMT+0530 (I...'. Red arrows point to the 'Name' column, the 'Instances' column, and the 'Availability Zones' column.

Then

Successfully done

The screenshot shows a browser window with three tabs open. The first tab is 'Load balancer details | EC2' with the URL '0.0.0.0:107.65'. The second tab is 'Not secure webloadbalancer-1069926602.ap-south-1.elb.amazonaws.com'. The third tab is 'webloadbalancer-1069926601' with the URL 'webloadbalancer-1069926601'. The main content area of the browser has a red border and contains the text 'Hello from Auto Scaling!'. Red arrows point from the tabs to the main content area, and another red arrow points to the text 'Hello from Auto Scaling!'.

Output:-

* To check the load balancer :-

Go to **EC2** at left side scroll down till **Load balancer** then click created **Load Balancer (webloadbalancer)** then click on it , In Details see the DNS name Copy the **DNS link** and paste it to browser you can see on screen the file output.

The screenshot shows the AWS EC2 Load Balancers console. On the left, there's a sidebar with various navigation options like Lifecycle Manager, Network & Security, Load Balancing, Auto Scaling, and CloudShell. Under Load Balancing, a red arrow points to the 'Load Balancers' link. The main pane displays a single load balancer named 'webloadbalancer'. It shows details such as Load balancer type (Application), Status (Active), Scheme (Internet-facing), VPC (vpc-00b5915fbffe184b6), Availability Zones (subnet-0b22791683e2e7db6, subnet-0ed045f4eb1aeeb52, subnet-0ed045f4eb1aeeb52), and Date created (July 27, 2025). A red box highlights the 'DNS name info' section, which lists 'webloadbalancer-1069926602.ap-south-1.elb.amazonaws.com (A Record)'. Below this, there's a note about activating Windows.

&

The screenshot shows the AWS EC2 Instances console. On the left, there's a sidebar with options like Dashboard, EC2 Global View, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, and Elastic Block Store. Under Instances, a red arrow points to the 'main' instance in the list. The main pane shows two instances: 'webserver' (Instance ID i-09a5dfe049b45414, Running, t2.micro, 2/2 checks passed) and 'main' (Instance ID i-046df700e0bbd7255, Running, t2.micro, 2/2 checks passed). Below the table, there's a 'Select an instance' dropdown and a 'DNS browser' button, both highlighted with red boxes.

Then

*** To check the Auto Scaling :-**

Go to running **instances** select it then, in instance state click it there select the option **terminated**, click it then the running & selected instances will be **deleted** .

Note :- If I delete the running instances then also the browser page run .

The website does not go down even when one server delete.

Auto scaling+ load balancer keep our application available.

To delete all instances which ASG (Auto Scaling Group) manage

Go to auto scaling group click it then click on created webASG then go to action option select delete option and click it .

Do same for target group, load balancer, instances.

Task 4 :- ◆ IAM USER with CLI only access

Search IAM and enter

Then

Go to left side and click on User then Click on **Create User**

The screenshot shows the AWS IAM service interface. On the left, there's a navigation sidebar with 'Access management' expanded, showing 'Users' (which has a red arrow pointing to it). In the main area, there's a table titled 'Users (1) Info' with one entry: 'sam'. The 'Create user' button is highlighted with a red arrow.

Then

In Specify user details in user details

Name :- **CL IUser**

Then click on **Next** button

This screenshot shows the 'Specify user details' step of the 'Create user' wizard. It includes a sidebar with 'Step 1: Specify user details' selected. The main form has a 'User details' section with a 'User name' field containing 'CL IUser'. A note below the field says: 'The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = . @ _ - (hyphen)'. There are also optional checkboxes for AWS Management Console access and programmatic access. The 'Next' button is highlighted with a red arrow.

Then

In Set permission

At permission option select 3rd option **Attach policies directly**

Then

In **Permission policies**

Select the option of **AdministratorAccess**

Then click on **Next** button

Step 1
● Specify user details
Step 2
● Set permissions
Step 3
○ Review and create

Set permissions
Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

Permissions options

Add user to group
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

Copy permissions
Copy all group memberships, attached managed policies, and inline policies from an existing user.

Attach policies directly
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

Permissions policies (1/1383)
Choose one or more policies to attach to your new user.

| Policy name | Type | Attached entities |
|--|----------------------------|-------------------|
| AdministratorAccess | AWS managed - job function | 1 |
| AdministratorAccess-Amplify | AWS managed | 0 |
| AdministratorAccess-AWSElast... | AWS managed | 0 |
| AIOpsAssistantPolicy | AWS managed | 0 |
| AIOpsOperatorAccess | AWS managed | 0 |
| AIOpsReadOnlyAccess | AWS managed | 0 |
| AlexaForBusinessDeviceSetup | AWS managed | 0 |
| AlexaForBusinessFullAccess | AWS managed | 0 |
| AlexaForBusinessGatewayExec... | AWS managed | 0 |
| AlexaForBusinessLifesizeDeleg... | AWS managed | 0 |
| AlexaForBusinessNetworkProfi... | AWS managed | 0 |
| AlexaForBusinessPolyDelegate... | AWS managed | 0 |
| AlexaForBusinessReadOnlyAcc... | AWS managed | 0 |
| AmazonAPIGatewayAdminist... | AWS managed | 0 |
| AmazonAPIGatewayInvokeFull... | AWS managed | 0 |
| AmazonAPIGatewayPushToClio... | AWS managed | 0 |
| AmazonAppFlowFullAccess | AWS managed | 0 |
| AmazonAppFlowReadOnlyAccess | AWS managed | 0 |

▶ Set permissions boundary - optional

Cancel Previous Next

Then

In Review & create click the button at the bottom of Create user

Step 1
● Specify user details
Step 2
● Set permissions
Step 3
● Review and create

Review and create
Review your choices. After you create the user, you can view and download the autogenerated password, if enabled.

User details

User name: CLUser Console password type: None Require password reset: No

Permissions summary

| Name | Type | Used as |
|---------------------|----------------------------|--------------------|
| AdministratorAccess | AWS managed - job function | Permissions policy |

Tags - optional
Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.
No tags associated with the resource.

Add new tag
You can add up to 50 more tags.

Create user

Then

The screenshot shows the AWS IAM 'Users' page. A green success message at the top states: 'User created successfully. You can view and download the user's password and email instructions for signing in to the AWS Management Console.' Below this, the 'Users (2) Info' section displays two users: 'CLIUser' and 'sam'. The 'CLIUser' row has a 'Create access key' button. The 'sam' row has a 'Delete' button. The left sidebar includes sections for 'Access management' (User groups, Roles, Policies, Identity providers, Account settings, Root access management), 'Access reports' (Access Analyzer, Resource analysis, Unused access, Analyzer settings, Credential report, Organization activity, Service control policies), and links to CloudShell and Feedback.

Then

Click on created user ([CLIUser](#))

Then go to security credential option click it then scroll down till access key

Then click on Access key then create access key

In use case select CLI 1st option

Then at bottom select the box of understand

Then click next button

Then at set description tag skip it & click on button Create access key

The screenshot shows the 'Create access key' wizard, Step 3: 'Retrieve access keys'. It displays a success message: 'Access key created. This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.' It shows two access keys: 'AKIAUPMYS3Y26YDAKQHB' and a redacted 'Secret access key'. The 'Access key best practices' section lists: 'Never store your access key in plain text, in a code repository, or in code.', 'Disable or delete access key when no longer needed.', 'Enable least-privilege permissions.', and 'Rotate access keys regularly.' At the bottom are 'Download .csv file' and 'Done' buttons.

Above key deleted so make new key

New access key created

The screenshot shows the AWS IAM 'Create access key' page. A green success message box at the top says 'Access key created' with the note: 'This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.' Below this, there's a 'Retrieve access keys' section with a table showing one access key entry:

| Access key | Secret access key | Actions |
|----------------------|--|----------------------|
| AKIAUPMYM3Y2SRXTK73Z | N2NhIog/wCv/hvaWrHMxsXm8PPhY30ZzjeA176wT | Hide |

On the left, a sidebar lists steps: 'Access key best practices & alternatives', 'Step 2 - optional', 'Set description tag', 'Step 3', and 'Retrieve access keys' (which is selected). At the bottom, there are links for 'CloudShell', 'Feedback', and 'Cookie preferences'.

Then

Download the .csv file

Then click on **Done** button

The secret key is one time show means one time only it will display after that no.

Go to cmd and open it

Then

Write command :-

aws --version

```
\Users\Samarth Lasure>aws --version
aws-cli/2.27.57 Python/3.13.4 Windows/10 exe/AMD64
C:\Users\Samarth Lasure>
```

Then

aws configure --profile CLIUser

This we will login in CLIUser

```
C:\Users\Samarth Lasure>aws configure --profile CLIUser
AWS Access Key ID [None]: AKIAUPMYM3Y2SRXTK73Z
AWS Secret Access Key [None]: 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
Default region name [None]: ap-south-1
Default output format [None]: json
```

Then

aws configure list --profile CLIUser

this will list all profile details like access key, secret key, region, etc.

```
C:\Users\Samarth Lasure>aws configure list --profile CLIUser
Name           Value        Type    Location
-----          ----
profile        CLIUser      manual   --profile
access_key     ****K73Z**** shared-credentials-file
secret_key     ****76wT**** shared-credentials-file
region         ap-south-1   config-file  ~/.aws/config
```

**Now we successfully completed the 4th task :-
IAM USER using CLI access**

----- **THANKYOU** -----