AWS Task-3

Task Description:

1. Create a S3 bucket, with no public access and upload files to the bucket & view the logs using cloudwatch for the uploaded files.

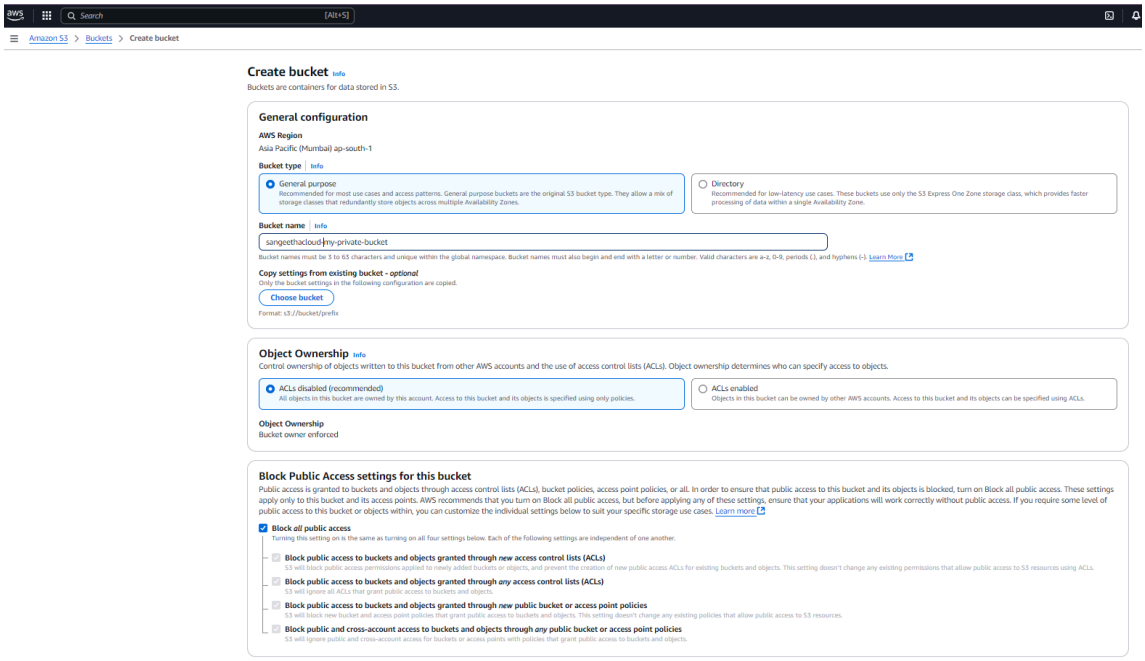
S3 Bucket with CloudWatch Logging - Step-by-Step Guide

Task Breakdown & Completion Status:

- Step 1: Created an S3 bucket with no public access.
- Step 2: Uploaded files to the private S3 bucket.
- Step 3: Enabled CloudTrail to track S3 activities.
- Step 4: Configure CloudWatch Logs to store and view logs.
- Step 5: Verified file upload (PutObject) and delete (DeleteObject) logs in CloudWatch.

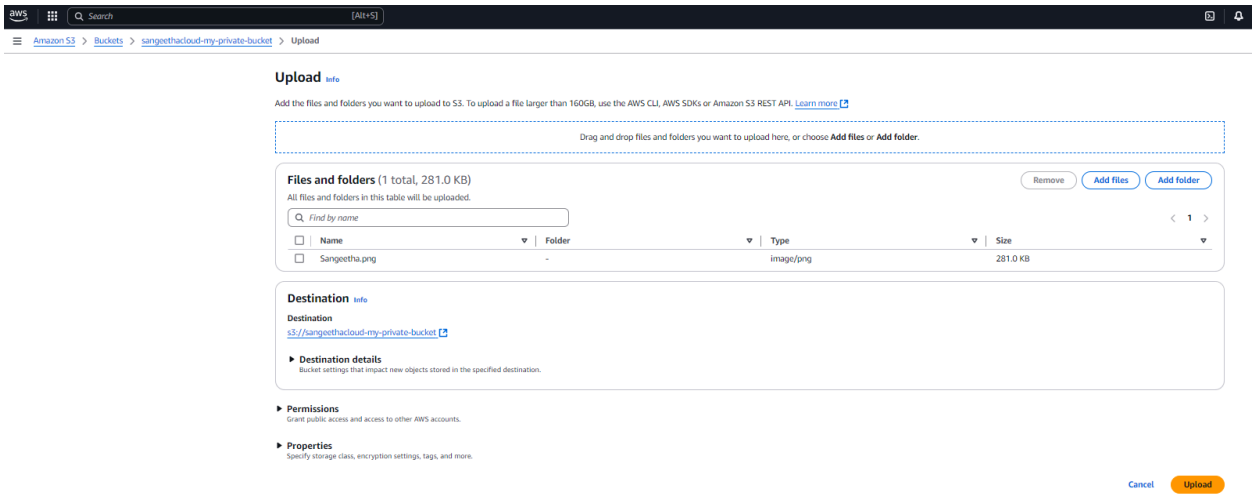
Step 1: Create an S3 Bucket (Private)

- - Login to AWS Console → Navigate to S3.
- - Click "Create bucket".
- - Enter a Bucket Name (sangeethacloud-my-private-bucket).
- - Choose a Region (same as your AWS resources).
- - Block all public access → Enable (this ensures the bucket remains private).
- - Object Ownership → Keep it ACLs disabled (recommended).
- - Click "Create bucket".
- - Bucket is now created with no public access.



Step 2: Upload Files to S3

- Open S3 → Click on your bucket (sangeethacloud-my-private-bucket).
- Click "Upload" → "Add files".
- Select a file from your system (e.g., an image or text file).
- Click "Upload".
- ✅ File is successfully uploaded to S3.

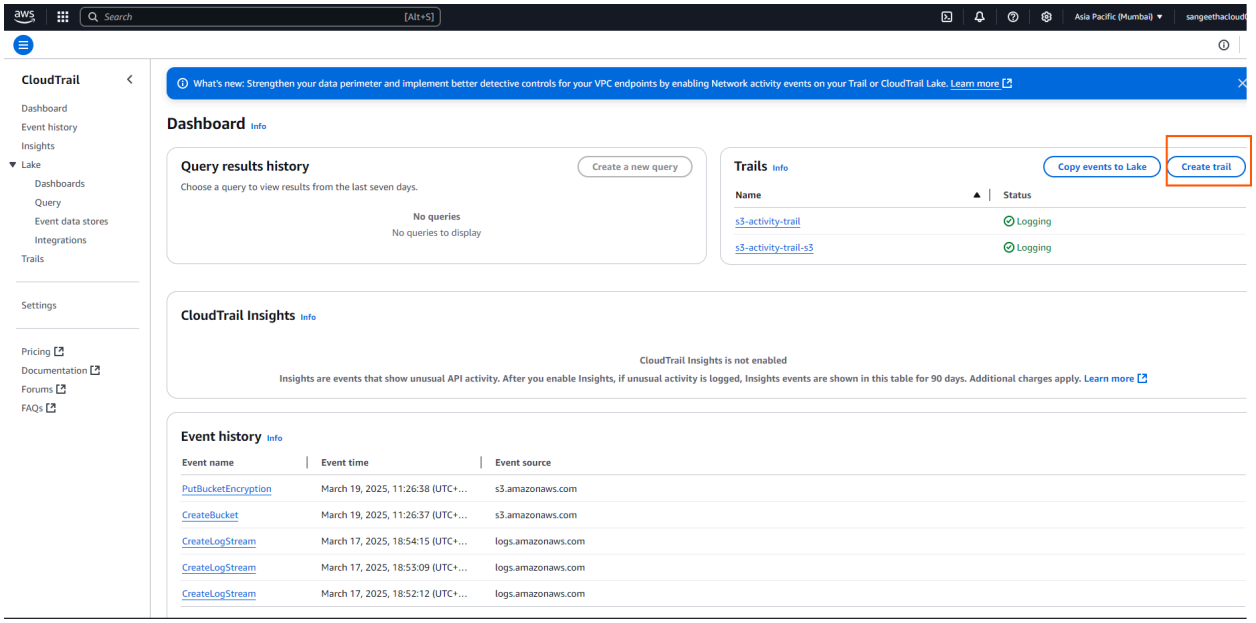


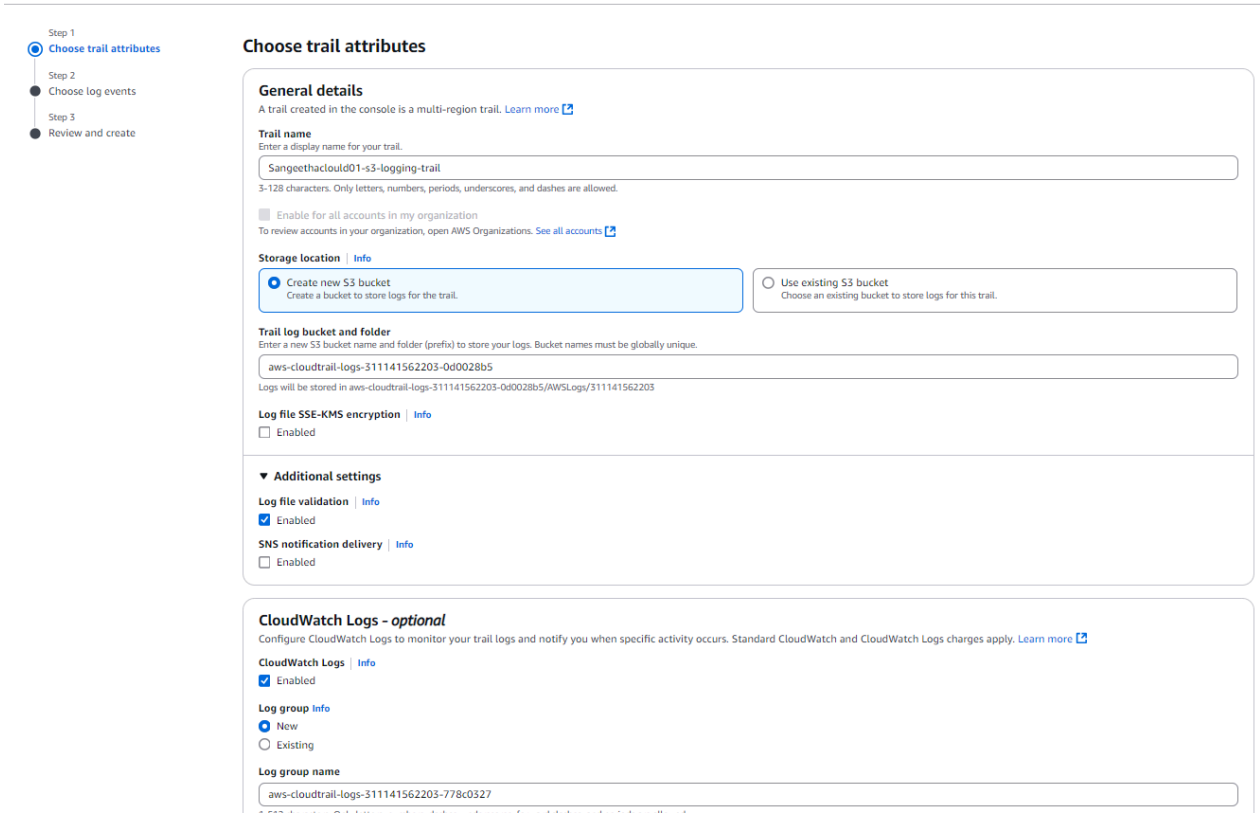
Step 3: Enable CloudTrail to Log S3 File Activities

To track uploads, downloads, and deletions, we need to enable CloudTrail.

- Go to AWS CloudTrail → Click "Create trail".
- Enter Trail Name: s3-logging-trail.
- Storage Location:
    - Select "Use existing S3 bucket" and choose sangeethacloud-my-private-bucket
    - This will store CloudTrail logs inside your bucket.

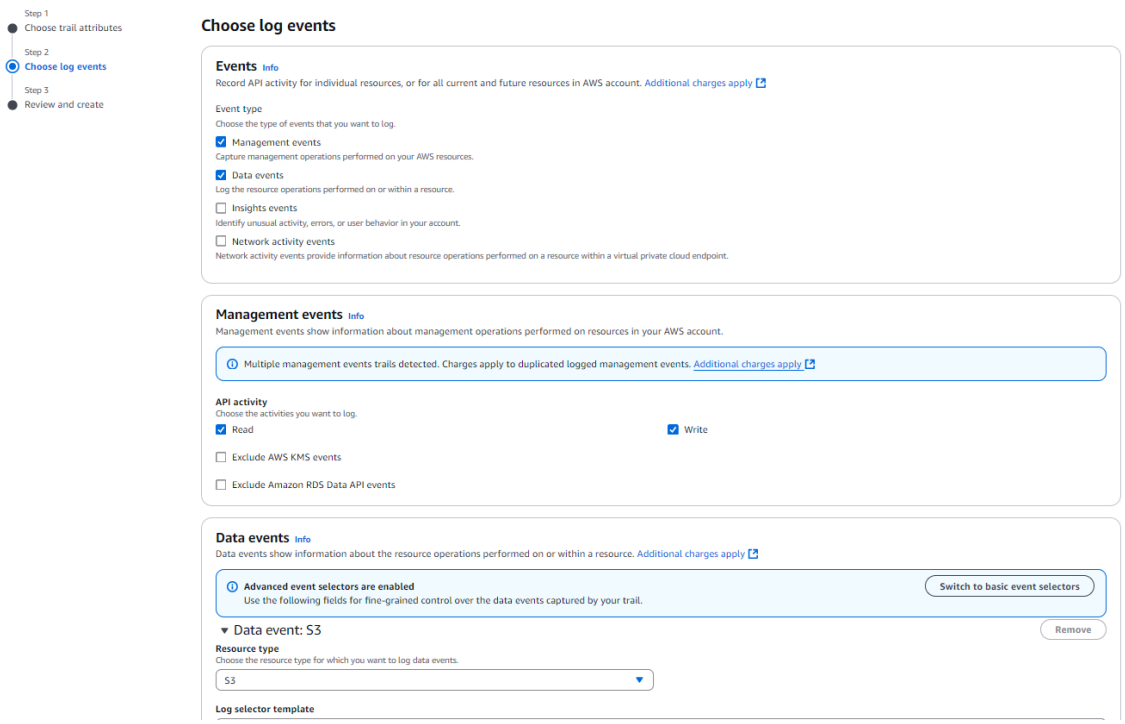
- create the tail



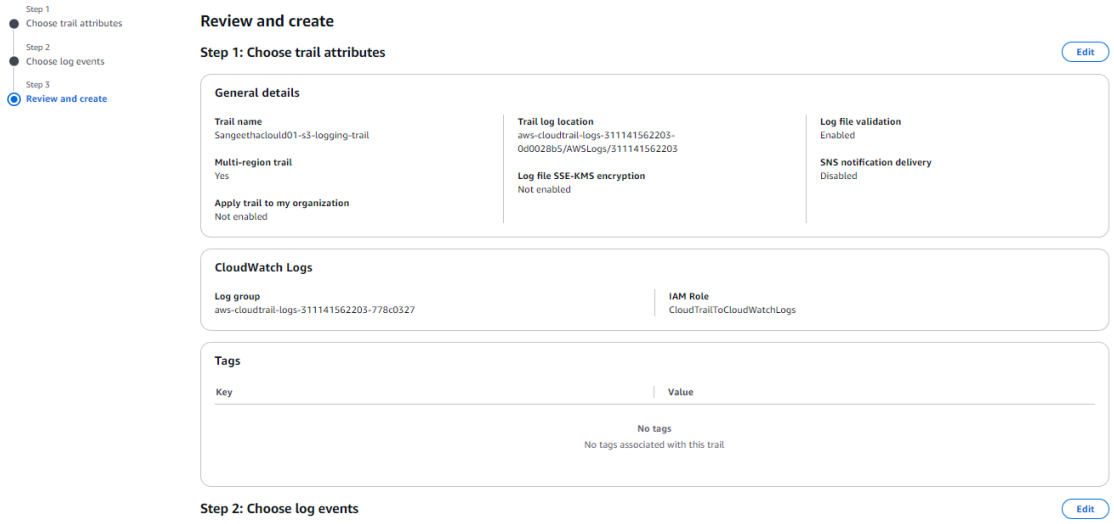


- **Enable Logging for S3:**
  - Scroll to Data events → Select S3.
  - Choose "Specify bucket" and select sangeethacloud-my-private-bucket.
  - Enable "Read & Write" (to log uploads & deletions).
- Click "Next", review the settings, and click "Create Trail".
- ✅ CloudTrail is now tracking S3 file activities.

Next step

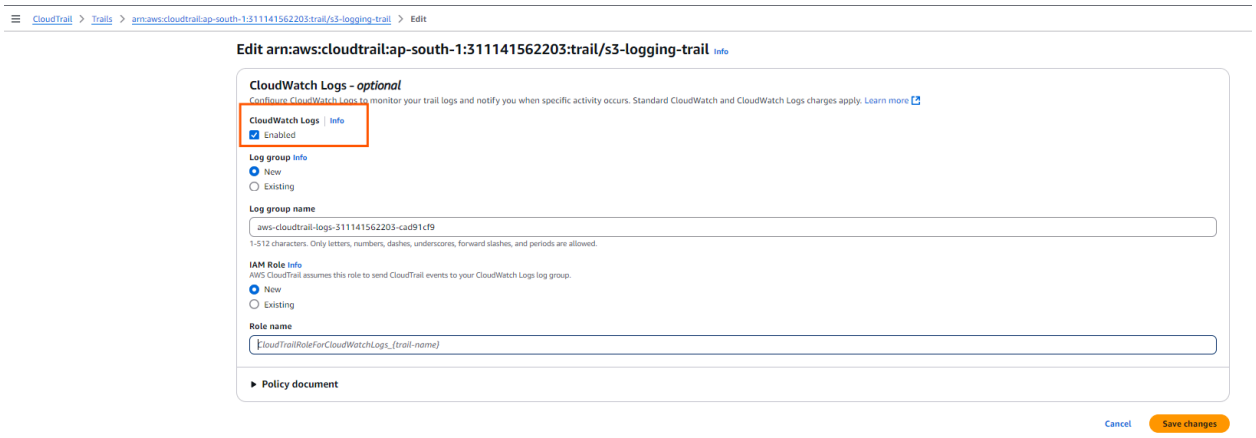


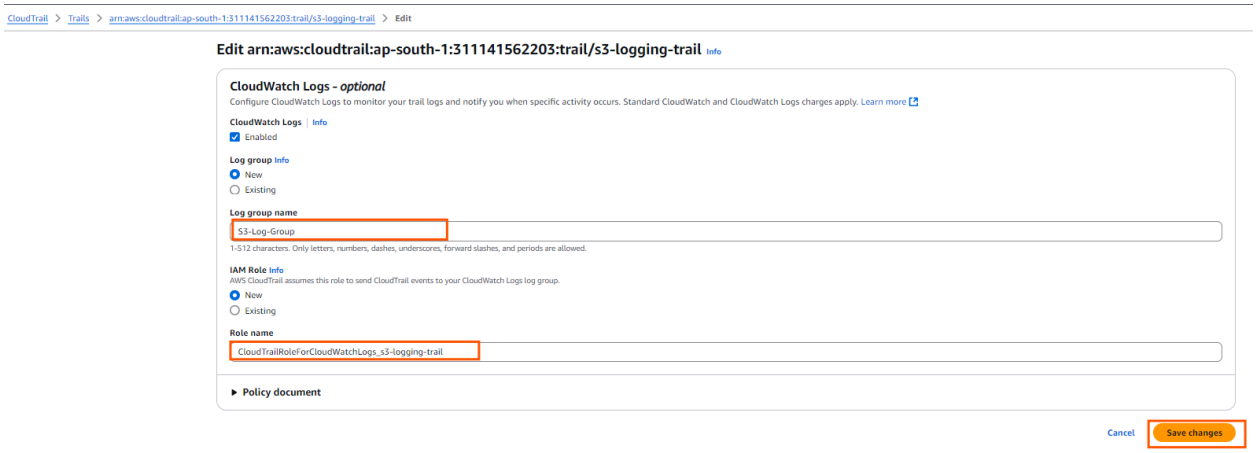
Review and proceed !



Step 4: Enable CloudWatch Logs for Easy Viewing

- Open AWS CloudTrail → Click on your trail (s3-logging-trail).
- Click "Edit" → Scroll to CloudWatch Logs.
- Enable CloudWatch Logs → Choose "Create new Log Group" (e.g., S3-Log-Group).
- Click "Save Changes".
- ✅ Now, S3 logs will be visible in CloudWatch Logs.





Step 5: View Logs in CloudWatch

- Go to AWS CloudWatch → Click "Log Groups".
- Open the Log Group (S3-Log-Group).
- Click on the latest Log Stream.
- Search for event types:
  - "PutObject" → When a file is uploaded
  - "DeleteObject" → When a file is deleted
  - "GetObject" → When a file is downloaded
- ✅ Now, you can track all file activities in CloudWatch.

aws Search [Alt+S] Asia Pacific (Mumbai) sangeethacloud

CloudWatch > Log groups > S3-Log-Group > 311141562203_CloudTrail_ap-south-1_4

CloudWatch

Favorites and recents

Dashboards

Alarms 0 0 0

Logs

Log groups New

Log Anomalies

Live Tail

Logs Insights New

Contributor Insights

Metrics

X-Ray traces New

Events

Application Signals

Network Monitoring New

Insights

Settings

Getting Started

What's new

Log events

Actions | Start tailing | Create metric filter

You can use the filter bar below to search for and match terms, phrases, or values in your log events. Learn more about filter patterns

DeleteObject

Clear 1m 30m 1h 12h Custom UTC timezone Display

Timestamp	Message
2025-03-19T06:33:16.913Z	{"eventVersion":"1.11","userIdentity":{"type":"Root","principalId":"311141562203","arn":"arn:aws:iam::311141562203:root","accountId":"311141562203","accessKeyId":"ASIAUQ4L3O5N7LWCIXNJ","userName":"sangeethacloud01","sessionContext":{"attributes":{"creationDate"…


```
{
    "eventVersion": "1.11",
    "userIdentity": {
        "type": "Root",
        "principalId": "311141562203",
        "arn": "arn:aws:iam::311141562203:root",
        "accountId": "311141562203",
        "accessKeyId": "ASIAUQ4L3O5N7LWCIXNJ",
        "userName": "sangeethacloud01",
        "sessionContext": {
            "attributes": {
                "creationDate": "2025-03-19T05:50:21Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2025-03-19T06:30:27Z",
    "eventSource": "s3.amazonaws.com",
    "eventName": "DeleteObjects",
    "awsRegion": "ap-south-1",
    "sourceIPAddress": "103.98.32.12",
    "userAgent": "[Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.0.0 Safari/537.36]",
    "requestParameters": {
        "bucketName": "sangeethacloud-my-private-bucket",
        "Host": "sangeethacloud-my-private-bucket.s3.ap-south-1.amazonaws.com",
        "delete": ""
    },
    "responseElements": null,
    "additionalEventData": {
        "SignatureVersion": "SigV4",
        "CipherSuite": "TLS_AES_128_GCM_SHA256",
        "bytesTransferredIn": 0,
        "AuthenticationMethod": "AuthHeader",
        "x-amz-id-2": "YlzP2L/0mdPWJ3QMEWxMUn6QZMsYArjNdQRWTpGJw+ITL2Czz3hw6jpqR42ALQnaCT79HGGBtg4bzDARV+pFFQ=",
        "bytesTransferredOut": 0
    },
    "requestID": "MZAB3CT094MEZ5TV",
    "eventID": "88beab00-7a5f-4dcf-87e5-69894091a61a",
    "readOnly": false,
    "resources": [
        {
            "type": "AWS::S3::Object",
            "ARNPrefix": "arn:aws:s3:::sangeethacloud-my-private-bucket/"
        },
        {
            "accountId": "311141562203",
            "type": "AWS::S3::Bucket",
            "ARN": "arn:aws:s3:::sangeethacloud-my-private-bucket"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": false,
```

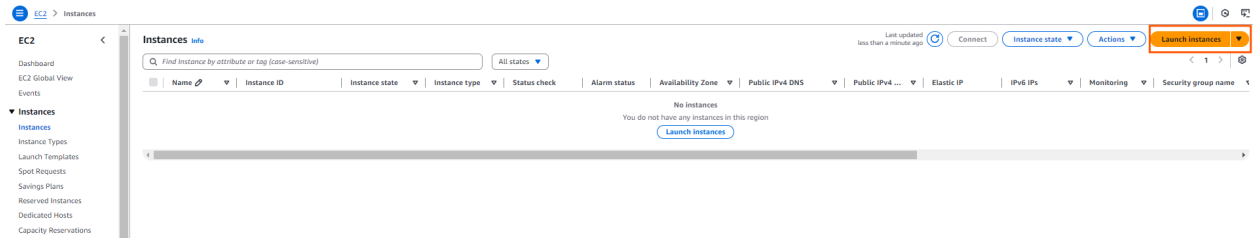
2 . Launch two ec2-instances and connect it to a application load balancer, where the output traffic from the server must be an load balancer IP address

1️⃣ Create Two EC2 Instances

1. Navigate to **AWS EC2 Console**.
2. Click **Launch Instance** and configure the following:
   - Choose **Amazon Linux 2** as the AMI.
   - Select **t2.micro** instance type.
   - Configure instance details and ensure both instances are in the same **VPC**.
   - Add **User Data** to install a web server:

     - #!/bin/bash
     - yum update -y
     - yum install -y httpd
     - echo "<h1>Welcome to Web Server 1</h1>" > /var/www/html/index.html
     - systemctl start httpd
     - systemctl enable httpd

   - Repeat the process for the second instance, but modify the HTML content:
     - echo "<h1>Welcome to Web Server 2</h1>" > /var/www/html/index.html
3. **Configure Security Group:**
   - Allow **HTTP (Port 80) from Anywhere (0.0.0.0/0)**.
   - Allow **SSH (Port 22) from Your IP**.
4. **Launch the Instances** and note their **Public IPs**.



It seems like you may be new to launching instances in EC2. Take a walkthrough to learn about EC2, how to launch instances and about best practices

Do not show me this message again | Take a walkthrough

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name

web-server-1

Add additional tags

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

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

Recents | Quick Start

Amazon Linux | macOS | Ubuntu | Windows | Red Hat | SUSE Linux | Debian

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

Amazon Machine Image (AMI)

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type — Free tier eligible
ami-08fe5144e4659a3b3 (64-bit (x86)) / ami-0bf9265d7d9594735 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Amazon Linux 2 comes with five years support. It provides Linux kernel 5.10 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras. This AMI is the successor of the Amazon Linux AMI that is now under maintenance only mode and has been removed from this wizard.

Amazon Linux 2 Kernel 5.10 AMI 2.0.20250305.0 x86_64 HVM gp2

Architecture	AMI ID	Publish Date	Username
64-bit (x86)	ami-08fe5144e4659a3b3	2025-03-05	ec2-user (Verified provider)

Summary

Number of instances | Info

1

Software Image (AMI)
Amazon Linux 2 Kernel 5.10 AMI...read more
ami-08fe5144e4659a3b3

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel | Launch instance | Preview code

Instance type Info | Get advice

Instance type

t2.micro — Free tier eligible
Family: t2 1 vCPU 1 GiB Memory Current generation: true On-Demand Linux base pricing: 0.0124 USD per Hour
On-Demand Windows base pricing: 0.017 USD per Hour On-Demand RHEL base pricing: 0.0268 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0142 USD per Hour On-Demand SUSE base pricing: 0.0124 USD per Hour

All generations

Compare instance types

Additional costs apply for AMIs with pre-installed software

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 - *required*

SangeethaCloud01

Create new key pair

Network settings Info

Edit

Network | Info

vpc-01118fb0ecfe7f2dc

Subnet | Info

No preference (Default subnet in any availability zone)

Auto-assign public IP | Info

Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups) | Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

- (●) Create security group
- ( ) Select existing security group

We'll create a new security group called '**launch-wizard-1**' with the following rules:

- [x] Allow SSH traffic from — Anywhere 0.0.0.0/0
  Helps you connect to your instance
- [ ] Allow HTTPS traffic from the internet
  To set up an endpoint, for example when creating a web server
- [x] Allow HTTP traffic from the internet
  To set up an endpoint, for example when creating a web server

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Create instance 2

It seeems like you may be new to launching instances in EC2. Take a walkthrough to learn about EC2, how to launch instances and about best practices

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name

web-server-2

Add additional tags

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

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

Recents | Quick Start

Amazon Linux | macOS | Ubuntu | Windows | Red Hat | SUSE Linux | Debian

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type
ami-08fe5144e4659a3b3 (64-bit (x86)) / ami-0bf9265d7d9594735 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible

Description

Amazon Linux 2 comes with five years support. It provides Linux kernel 5.10 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras. This AMI is the successor of the Amazon Linux AMI that is now under maintenance only mode and has been removed from this wizard.

Amazon Linux 2 Kernel 5.10 AMI 2.0.20250305.0 x86_64 HVM gp2

Architecture	AMI ID	Publish Date	Username	
64-bit (x86)	ami-08fe5144e4659a3b3	2025-03-05	ec2-user	Verified provider

Instance type Info | Get advice

Instance type

t2.micro Free tier eligible
Family: t2 1 vCPU 1 GiB Memory Current generation: true On-Demand Linux base pricing: 0.0124 USD per Hour
On-Demand Windows base pricing: 0.017 USD per Hour On-Demand RHEL base pricing: 0.0268 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0142 USD per Hour On-Demand SUSE base pricing: 0.0124 USD per Hour

All generations

Compare instance types

Additional costs apply for AMIs with pre-installed software

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 - *required*

SangeethaCloud01

Create new key pair

Network settings Info

Edit

Network | Info

vpc-01118fb0ecfe7f2dc

Subnet | Info

No preference (Default subnet in any availability zone)

Auto-assign public IP | Info

Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups) | Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

- (●) Create security group
- ( ) Select existing security group

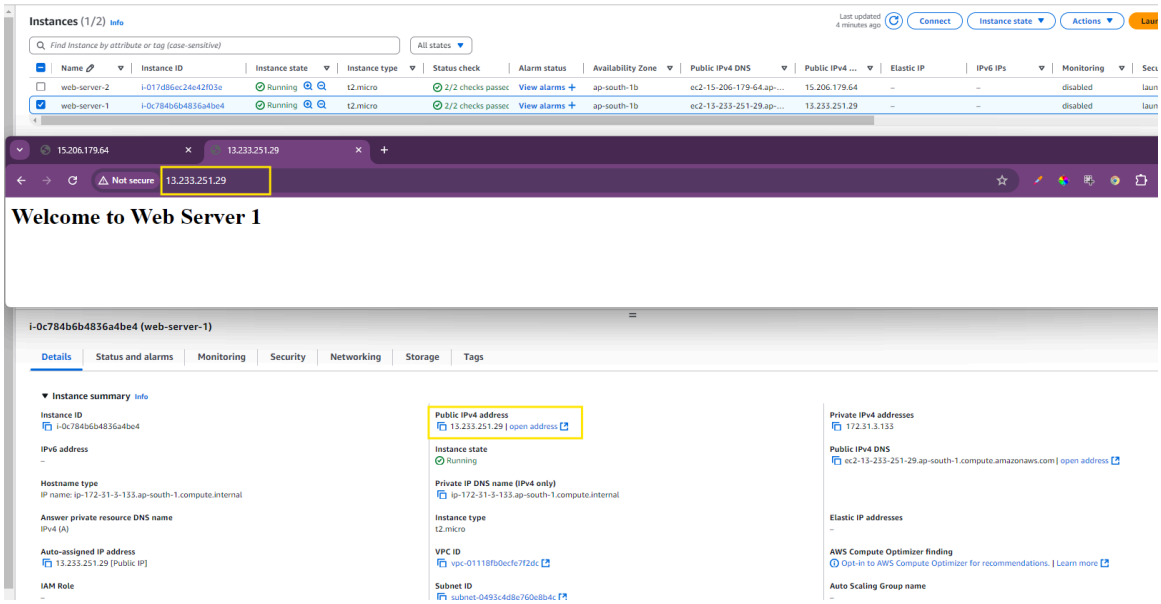
We'll create a new security group called '**launch-wizard-2**' with the following rules:

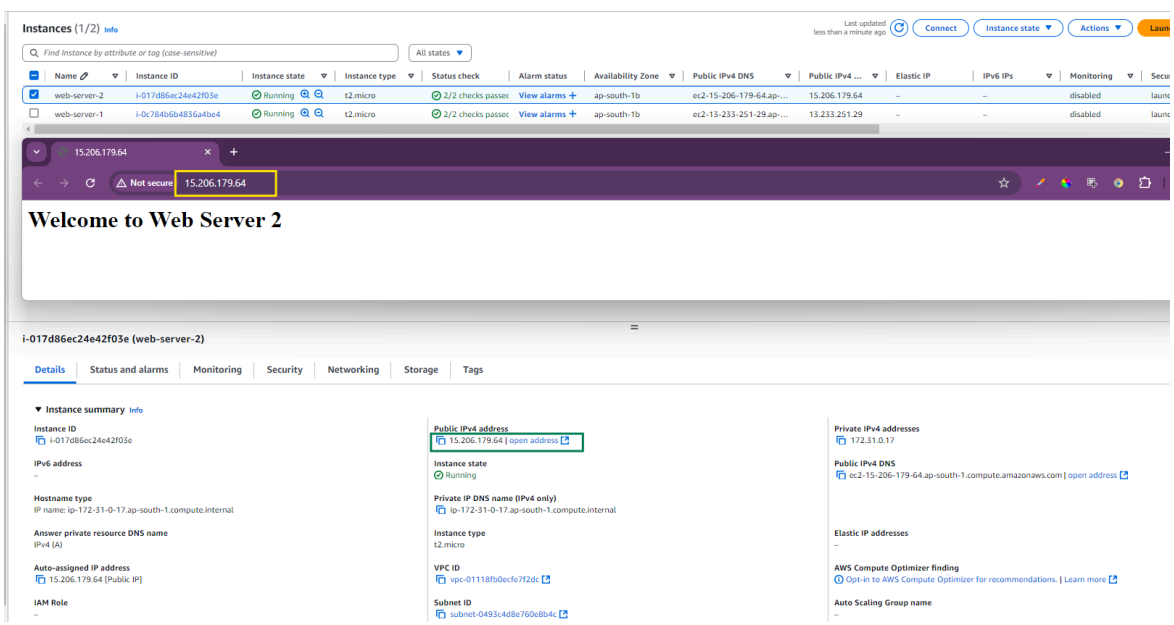
- [x] Allow SSH traffic from — Helps you connect to your instance — Anywhere 0.0.0.0/0
- [ ] Allow HTTPS traffic from the internet — To set up an endpoint, for example when creating a web server
- [x] Allow HTTP traffic from the internet — To set up an endpoint, for example when creating a web server

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

2 Verify Web Server is Running

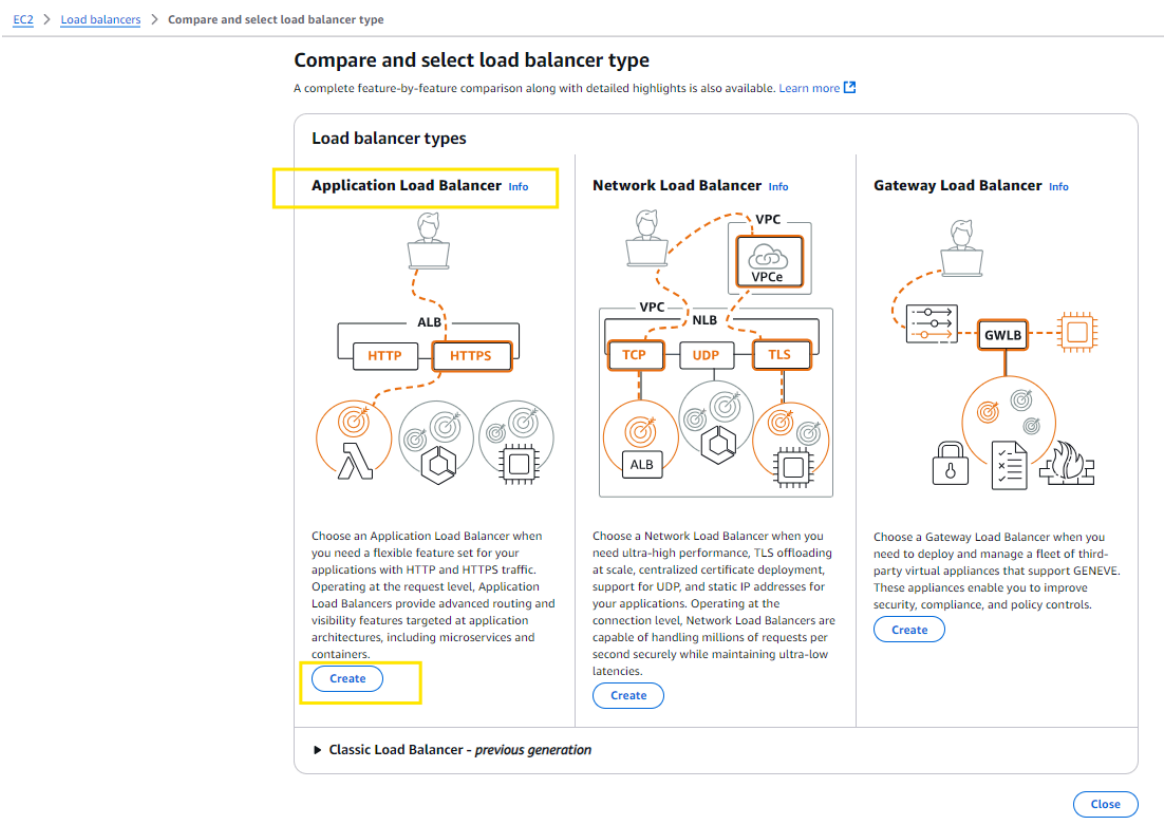
1. Connect to the instances using SSH:
   ssh -i "your-key.pem" ec2-user@<EC2-Public-IP>
2. Check if the web server is running:
   curl http://localhost
   Expected Output:
   <h1>Welcome to Web Server 1</h1> **OR** <h1>Welcome to Web Server 2</h1>
3. Open the **Public IP** in a browser and check if the webpage is displaying correctly.





3️⃣ Create an Application Load Balancer (ALB)

1. Go to **AWS Load Balancers Console**.
2. Click **Create Load Balancer** → Select **Application Load Balancer**.
3. Configure:
   - **Scheme**: Internet-facing.
   - **IP Address Type**: IPv4.
   - **Availability Zones**: Select at least **two subnets**.
4. **Configure Security Group:**
   - Allow **HTTP (Port 80) from Anywhere**.
5. **Create a Target Group:**
   - Choose **Instance Type** as target.
   - **Port:** 80.
   - **Health Check Path:** `/index.html` (change from `/` to avoid failure).
6. **Register Targets (EC2 Instances)**:
   - Select both instances and register them.

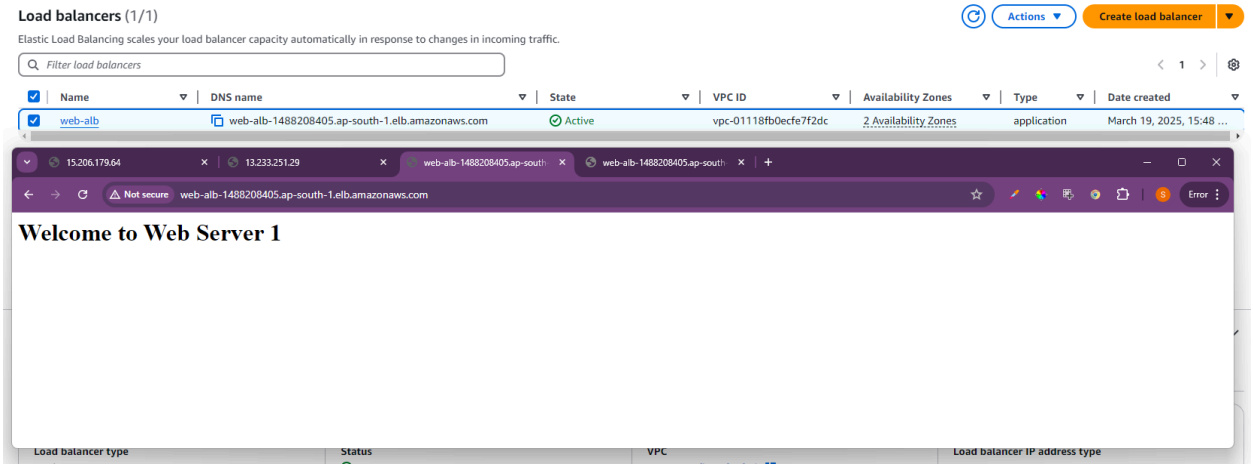


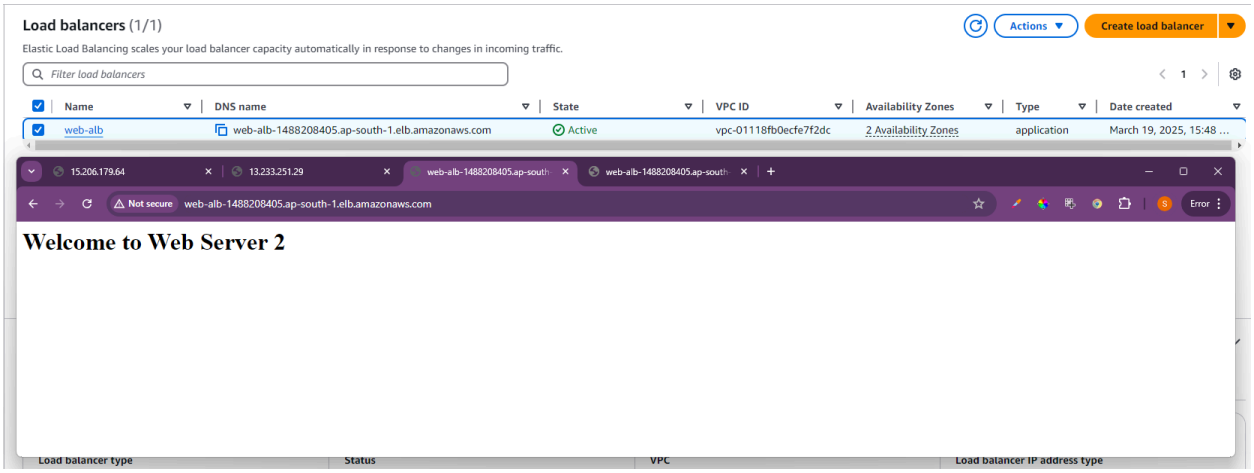
4️⃣ Testing the Load Balancer

1. Wait for the **ALB status to become Active**.
2. Open the **Load Balancer DNS Name** in a browser.
   – http://web-alb-xxxxxx.ap-south-1.elb.amazonaws.com/

3. **Expected Output:**

- Refresh multiple times, and you should see responses alternating between:
- Welcome to Web Server 1
- Welcome to Web Server 2





🎯 Final Confirmation

✅ Web Servers are running
✅ Load Balancer is distributing traffic
✅ DNS is accessible and responding correctly