SERVICES S3+EC2 +RDS

S3

1. Create S3 Bucket

1. Go to AWS S3 Console
2. Click **Create bucket**
3. Configure:
   - **Bucket name**: your-image-gallery-bucket
   - **AWS Region**: Same as your EC2/RDS
   - **Block Public Access**: *Uncheck all* (temporarily)
   - **Bucket Versioning**: Enable (recommended)
4. Click **Create**

2.UPLOAD FILES IN BUCKET

3. Configure Bucket Policy

1. Go to **Permissions → Bucket Policy**
2. Add this policy (replace YOUR-BUCKET-NAME):
3. GENERATE POLICY

json

Copy

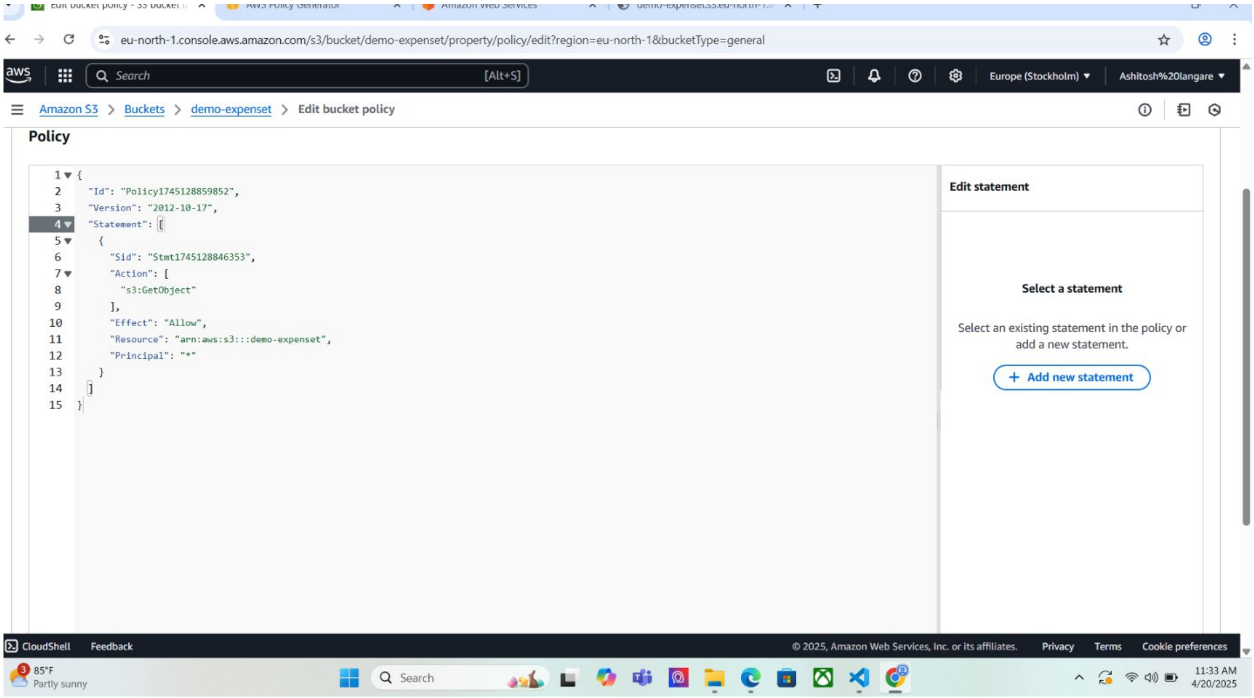
Download

{

SERVICES S3+EC2 +RDS

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": "*",
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Resource": "arn:aws:s3:::YOUR-BUCKET-NAME/*"
    }
  ]
}
```

3. Click **Save**



eu-north-1.console.aws.amazon.com/s3/bucket/demo-expenset/property/policy/edit?region=eu-north-1&bucketType=general

Amazon S3 > Buckets > demo-expenset > Edit bucket policy

Policy

```
{
  "Id": "Policy1745128859852",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1745128846353",
      "Action": [
        "s3:GetObject"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::demo-expenset",
      "Principal": "*"
    }
  ]
}
```

Edit statement

Select a statement

Select an existing statement in the policy or add a new statement.

4. Configure IAM Role (for EC2)

SERVICES S3+EC2 +RDS

1. Create IAM Role:
   - **Service**: EC2
   - Permissions: AmazonS3FullAccess (or create custom policy)
2. Attach role to EC2 instance:
   - EC2 Dashboard → Instances → Instance → Actions → Security → Modify IAM Role

WHY I USED S3 ?

✔ **Scalability** – Handles **petabytes of data** effortlessly.

✔ **Security** – Supports **encryption, access control, IAM roles** for protection.

✔ **Durability** – Designed for **99.999999999% durability** (data is highly redundant).

✔ **Cost Efficiency** – Choose between **Standard, IA, Glacier** storage tiers.

✔ **High Availability** – Data is accessible from anywhere via **S3 URLs or APIs**.

WHEN IT START CHARGING MONEY

Amazon S3 starts charging **as soon as you store data** in a bucket. However, AWS offers a **free tier** that provides **5GB of standard storage** for free each month. Once you exceed this limit, charges apply based on:

- **Storage usage** (GB stored per month).
- **Requests & data retrieval** (GET, PUT, LIST operations).
- **Data transfer** (moving data out of AWS).
- **Storage class**

SERVICES S3+EC2 +RDS

EC2

Common EC2 Use Cases

- ***Hosting Websites & Web Apps*** *(PHP, Node.js, Django, etc.).*
- ***Running Databases*** *(MySQL, PostgreSQL, MongoDB).*
- ***Machine Learning & AI Workloads*** *(GPU-powered instances).*

STEPS IN EC2

1. Launch EC2 Instance

1. *Go to [EC2 Dashboard]()*
2. *Click **Launch Instance***
3. *Configure:*
    - ***Name**: Web-Server*
    - ***AMI**: Amazon Linux 2023 or Ubuntu 22.04 LTS*
    - ***Instance Type**: t2.micro (free tier eligible)*
    - ***Key Pair**: Create/download new .pem key*
    - ***Security Group**:*
        - *Allow SSH (Port 22) - Restrict to your IP*
        - *Allow HTTP (Port 80)*

- *Allow HTTPS (Port 443)*

4. *Click **Launch***

2. Connect to Instance

bash

Copy

Download

```
# For Linux/Mac
chmod 400 your-key.pem
ssh -i "your-key.pem" ec2-user@your-instance-ip

# For Windows use PuTTY
```

3. Install Required Packages

bash

Copy

Download

```
# Update system
sudo apt update && sudo apt upgrade -y  # Ubuntu
sudo dnf update -y                      # Amazon Linux

# Install LAMP Stack
sudo apt install apache2 php php-mysql php-curl php-gd php-mbstring php-xml php-zip libapache2-mod-php -y

# Install AWS CLI (for S3 access)
```

```
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install
```

4. Configure Web Server

1. *Set document root permissions:*

bash

Copy

Download

```
sudo chown -R www-data:www-data /var/www/html
sudo chmod -R 755 /var/www/html
```

2. *Configure PHP (edit /etc/php/[version]/apache2/php.ini):*

ini

Copy

Download

```
upload_max_filesize = 64M
post_max_size = 64M
display_errors = Off
log_errors = On
```

3. *Restart Apache:*

bash

Copy

Download

```
sudo systemctl restart apache2
```

SERVICES S3+EC2 +RDS

5. Configure IAM Role (for S3 Access)

1. *Create IAM Role with policy:*

json

Copy

Download

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::your-bucket-name",
        "arn:aws:s3:::your-bucket-name/*"
      ]
    }
  ]
}
```

2. *Attach role to EC2 instance:*
   - *EC2 Dashboard → Instances → Actions → Security → Modify IAM Role*

7. Enable HTTPS (SSL)

SERVICES S3+EC2 +RDS

1. *Install Certbot:*

bash

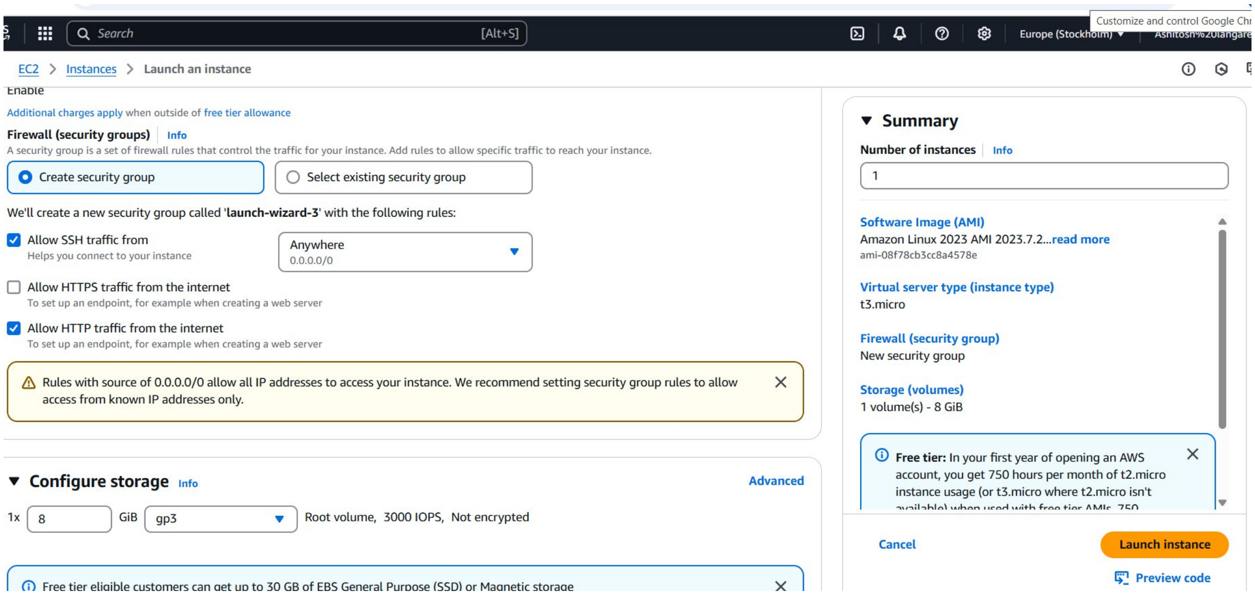
Copy

Download

sudo snap install --classic certbot

sudo certbot --apache

2. *Follow prompts to configure SSL certificate*


Search [Alt+S]
Europe (Stockholm) ▾
Customize and control Google Chr
EC2 > Instances > Launch an instance

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-3**' with the following rules:

☑ 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

☑ 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.

▼ Configure storage Info — Advanced

1x 8 GiB gp3 Root volume, 3000 IOPS, Not encrypted

ⓘ Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage

▼ Summary

Number of instances Info

1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.7.2...read more
ami-08f78cb3cc8a4578e

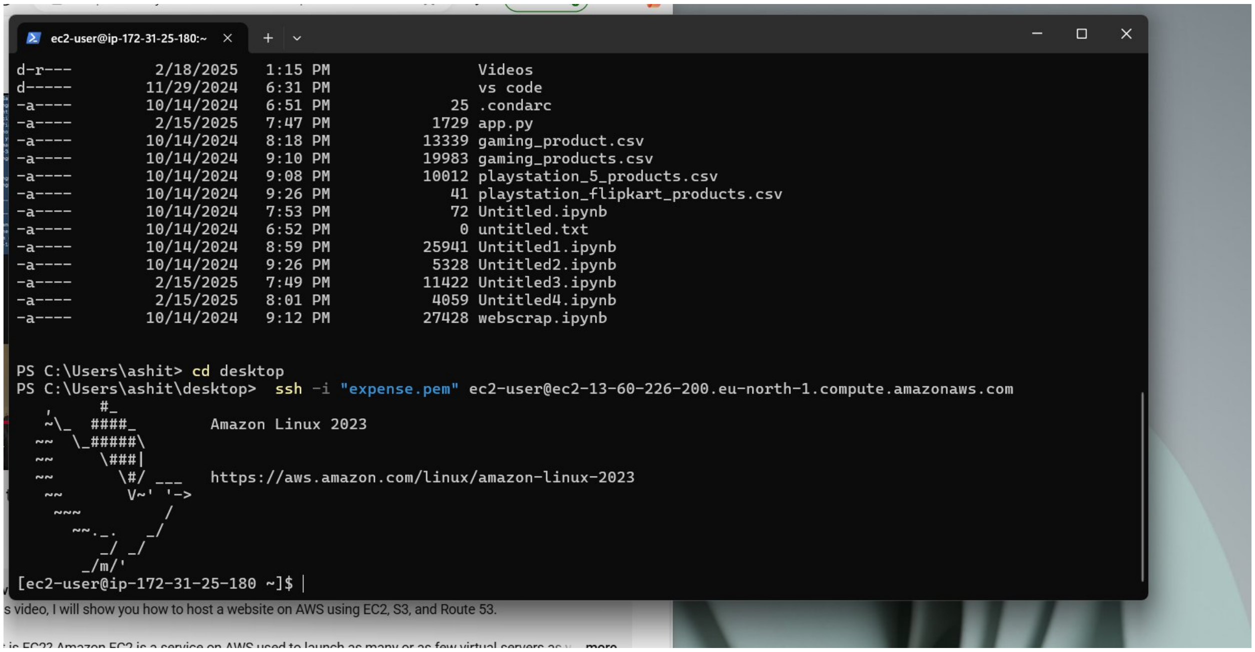
Virtual server type (instance type)
t3.micro

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

SERVICES S3+EC2 +RDS

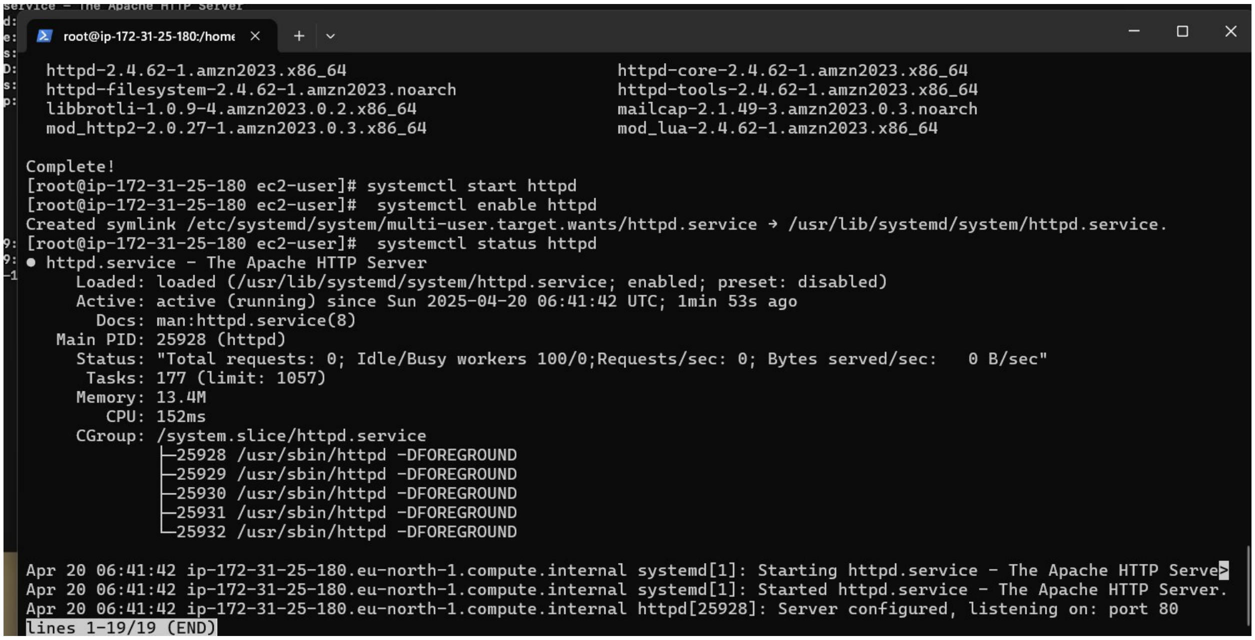


```
d-r---         2/18/2025   1:15 PM                Videos
d-----        11/29/2024   6:31 PM                vs code
-a----        10/14/2024   6:51 PM             25 .condarc
-a----         2/15/2025   7:47 PM           1729 app.py
-a----        10/14/2024   8:18 PM          13339 gaming_product.csv
-a----        10/14/2024   9:10 PM          19983 gaming_products.csv
-a----        10/14/2024   9:08 PM          10012 playstation_5_products.csv
-a----        10/14/2024   9:26 PM             41 playstation_flipkart_products.csv
-a----        10/14/2024   7:53 PM             72 Untitled.ipynb
-a----        10/14/2024   6:52 PM              0 untitled.txt
-a----        10/14/2024   8:59 PM          25941 Untitled1.ipynb
-a----        10/14/2024   9:26 PM           5328 Untitled2.ipynb
-a----         2/15/2025   7:49 PM          11422 Untitled3.ipynb
-a----         2/15/2025   8:01 PM           4059 Untitled4.ipynb
-a----        10/14/2024   9:12 PM          27428 webscrap.ipynb


PS C:\Users\ashit> cd desktop
PS C:\Users\ashit\desktop>  ssh -i "expense.pem" ec2-user@ec2-13-60-226-200.eu-north-1.compute.amazonaws.com
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-25-180 ~]$ |
```

s video, I will show you how to host a website on AWS using EC2, S3, and Route 53.

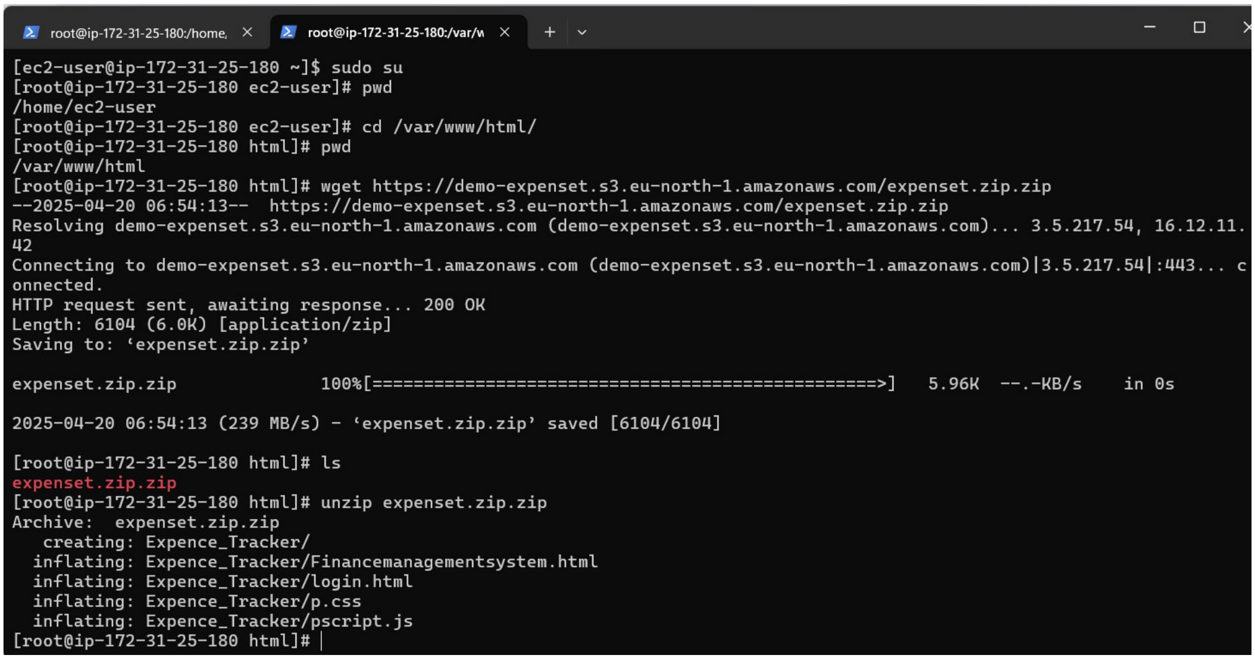
is EC2? Amazon EC2 is a service on AWS used to launch as many or as few virtual servers as y... more



```
  httpd-2.4.62-1.amzn2023.x86_64                        httpd-core-2.4.62-1.amzn2023.x86_64
  httpd-filesystem-2.4.62-1.amzn2023.noarch             httpd-tools-2.4.62-1.amzn2023.x86_64
  libbrotli-1.0.9-4.amzn2023.0.2.x86_64                 mailcap-2.1.49-3.amzn2023.0.3.noarch
  mod_http2-2.0.27-1.amzn2023.0.3.x86_64                mod_lua-2.4.62-1.amzn2023.x86_64

Complete!
[root@ip-172-31-25-180 ec2-user]# systemctl start httpd
[root@ip-172-31-25-180 ec2-user]#  systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[root@ip-172-31-25-180 ec2-user]#  systemctl status httpd
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
     Active: active (running) since Sun 2025-04-20 06:41:42 UTC; 1min 53s ago
       Docs: man:httpd.service(8)
   Main PID: 25928 (httpd)
     Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec:   0 B/sec"
      Tasks: 177 (limit: 1057)
     Memory: 13.4M
        CPU: 152ms
     CGroup: /system.slice/httpd.service
             ├─25928 /usr/sbin/httpd -DFOREGROUND
             ├─25929 /usr/sbin/httpd -DFOREGROUND
             ├─25930 /usr/sbin/httpd -DFOREGROUND
             ├─25931 /usr/sbin/httpd -DFOREGROUND
             └─25932 /usr/sbin/httpd -DFOREGROUND

Apr 20 06:41:42 ip-172-31-25-180.eu-north-1.compute.internal systemd[1]: Starting httpd.service - The Apache HTTP Serve>
Apr 20 06:41:42 ip-172-31-25-180.eu-north-1.compute.internal systemd[1]: Started httpd.service - The Apache HTTP Server.
Apr 20 06:41:42 ip-172-31-25-180.eu-north-1.compute.internal httpd[25928]: Server configured, listening on: port 80
lines 1-19/19 (END)
```



```
[ec2-user@ip-172-31-25-180 ~]$ sudo su
[root@ip-172-31-25-180 ec2-user]# pwd
/home/ec2-user
[root@ip-172-31-25-180 ec2-user]# cd /var/www/html/
[root@ip-172-31-25-180 html]# pwd
/var/www/html
[root@ip-172-31-25-180 html]# wget https://demo-expenset.s3.eu-north-1.amazonaws.com/expenset.zip.zip
--2025-04-20 06:54:13--  https://demo-expenset.s3.eu-north-1.amazonaws.com/expenset.zip.zip
Resolving demo-expenset.s3.eu-north-1.amazonaws.com (demo-expenset.s3.eu-north-1.amazonaws.com)... 3.5.217.54, 16.12.11.
42
Connecting to demo-expenset.s3.eu-north-1.amazonaws.com (demo-expenset.s3.eu-north-1.amazonaws.com)|3.5.217.54|:443... c
onnected.
HTTP request sent, awaiting response... 200 OK
Length: 6104 (6.0K) [application/zip]
Saving to: ‘expenset.zip.zip’

expenset.zip.zip             100%[==================================================>]   5.96K  --.-KB/s    in 0s

2025-04-20 06:54:13 (239 MB/s) - ‘expenset.zip.zip’ saved [6104/6104]

[root@ip-172-31-25-180 html]# ls
expenset.zip.zip
[root@ip-172-31-25-180 html]# unzip expenset.zip.zip
Archive:  expenset.zip.zip
   creating: Expence_Tracker/
  inflating: Expence_Tracker/Financemanagementsystem.html
  inflating: Expence_Tracker/login.html
  inflating: Expence_Tracker/p.css
  inflating: Expence_Tracker/pscript.js
[root@ip-172-31-25-180 html]# |
```

RDS

Common Use Cases

- ***Web Applications*** *– Store user data, authentication details, and transactions.*
- ***Enterprise Databases*** *– Manage large-scale business data securely.*
- ***E-commerce Platforms*** *– Handle product catalogs, orders, and customer details.*
- ***Analytics & Reporting*** *– Process large datasets efficiently.*

STEPS IN RDS

Step 1: Access the RDS Dashboard

1. *Log in to your **AWS Management Console**.*

2. *Navigate to **Amazon RDS** from the AWS services menu.*

Step 2: Choose a Database Engine

1. *Click **Create Database**.*
2. *Select a **database engine** (MySQL, PostgreSQL, MariaDB, SQL Server, or Oracle).*
3. *Choose **Standard database creation** for full customization.*

Step 3: Configure Database Settings

1. *Set a **DB instance identifier** (unique name for your database).*
2. *Choose a **username & password** for database access.*
3. *Select **Instance type** (based on CPU, memory, and performance needs).*

Step 4: Configure Storage & Backup

1. *Select **storage type** (General Purpose SSD, Provisioned IOPS, or Magnetic).*
2. *Enable **automatic backups** for data recovery.*
3. *Set **retention period** for backups.*

Step 5: Set Up Security & Networking

1. *Choose a **VPC & Subnet group** for network access.*
2. *Configure **Security Groups** to allow access from EC2 or other services.*
3. *Enable **IAM authentication** for secure access.*

Step 6: Finalize & Create the Database

1. *Review all settings.*
2. *Click **Create Database**.*
3. *Wait for the instance to be provisioned.*

Step 7: Connect to Your RDS Database

1. *Find your **RDS endpoint** in the AWS console.*
2. *Use a database client (MySQL Workbench, pgAdmin, etc.) or connect via EC2*

WHEN THIS SERVICES START CHARGING MONEY

*AWS services like **RDS, S3, and EC2** start charging money based on usage. However, AWS offers a **Free Tier** for new users, which provides limited free usage for the first **12 months**. Here's when each service starts charging:*

Amazon EC2 Pricing

- ***Free Tier**: 750 hours per month of **t2.micro** or **t3.micro** instances (Linux/Windows).*
- ***Charges Start**: Once you exceed **750 hours/month** or use larger instance types.*
- ***Billing Model**: Pay-as-you-go (per second/hour), Reserved Instances (discounted long-term), or Spot Instances (cheaper, but variable availability).*

Amazon S3 Pricing

- ***Free Tier**: 5GB of **Standard Storage**, 20,000 GET requests, 2,000 PUT requests per month.*
- ***Charges Start**: When storage exceeds **5GB**, or if you use **Glacier, Intelligent-Tiering, or frequent data transfers**.*
- ***Billing Model**: Charged per GB stored, retrieval requests, and data transfer.*

Amazon RDS Pricing

- ***Free Tier**: 750 hours per month of **Single-AZ db.t2.micro** for MySQL, PostgreSQL, MariaDB.*
- ***Charges Start**: When you exceed **750 hours/month**, use Multi-AZ, or larger instance types.*
- ***Billing Model**: Pay-as-you-go (per hour), Reserved Instances (discounted long-term), and storage costs.*