

## Task-4.

### Application Access

Access app via:

- o Elastic IP or
- o EC2 Public IP
- o (Optional) Route 53 domain

step 1: Launch and connect to task-4 instance.

The screenshot shows the AWS EC2 Instances page. A green notification bar at the top indicates "Successfully initiated stopping of i-0977102f84ad5cc28". The main table lists one instance: "task-4" (Instance ID: i-0fe3ecdfbdcc54d7a, Instance state: Running, Instance type: t2.micro, Status check: Initializing). The "Actions" dropdown menu for this instance includes options like "Stop", "Start", "Reboot", "Terminate", and "Launch instances". On the left sidebar, under the "Instances" section, "Capacity Manager" is highlighted. The "Details" tab is selected for the instance "i-0977102f84ad5cc28 (task-4)". The "Public IPv4 address" is listed as 65.240.99, and the "Private IPv4 addresses" are listed as 172.31.1.172 and ec2-65-2-40-99.ap-south-1.compute.amazonaws.com.

The screenshot shows the "EC2 Instance Connect" dialog box. It has tabs for "EC2 Instance Connect", "SSM Session Manager", "SSH client", and "EC2 serial console". The "EC2 Instance Connect" tab is active. The "Instance ID" field shows "i-0fe3ecdfbdcc54d7a (task-4)". Under "Connection type", the "Connect using a Public IP" option is selected, with the IP address 13.127.47.232. There is also an option for "Connect using a Private IP". The "Username" field contains "ec2-user". A note at the bottom states: "Note: In most cases, the default username, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username." At the bottom right are "Cancel" and "Connect" buttons.

## Step 2: Creating Simple Application Using HTML so for that installing httpd

Sudo yum install httpd -y

```
[root@ip-172-31-40-21 ~]# yum install httpd -y
Last metadata expiration check: 2:14:03 ago on Sat Feb 7 17:48:16 2026.
Dependencies resolved.
=====
Package          Architecture Version       Repository  Size
=====
Installing:
httpd           x86_64      2.4.66-1.amzn2023.0.1   amazonlinux 47 k
Installing dependencies:
apr              x86_64      1.7.5-1.amzn2023.0.4   amazonlinux 129 k
apr-util         x86_64      1.6.3-1.amzn2023.0.2   amazonlinux 97 k
apr-util-lmdb    x86_64      1.6.3-1.amzn2023.0.2   amazonlinux 13 k
generic-logos-httpd noarch      18.0.0-12.amzn2023.0.3  amazonlinux 19 k
httpd-core       x86_64      2.4.66-1.amzn2023.0.1   amazonlinux 1.4 M
httpd-filesystem noarch      2.4.66-1.amzn2023.0.1   amazonlinux 13 k
httpd-tools      x86_64      2.4.66-1.amzn2023.0.1   amazonlinux 81 k
libbrotli        x86_64      1.0.9-4.amzn2023.0.2   amazonlinux 315 k
mailcap          noarch      2.1.49-3.amzn2023.0.3  amazonlinux 33 k
Installing weak dependencies:
apr-util-openssl x86_64      1.6.3-1.amzn2023.0.2   amazonlinux 15 k
mod_http2        x86_64      2.0.27-1.amzn2023.0.3  amazonlinux 166 k
mod_uua          x86_64      2.4.66-1.amzn2023.0.1   amazonlinux 60 k
=====
Transaction Summary
=====
Install 13 Packages

Total download size: 2.4 M
Installed size: 6.9 M
Downloading Packages:
(1/13): apr-1.7.5-1.amzn2023.0.4.x86_64.rpm 3.7 MB/s | 129 kB 00:00
(2/13): apr-util-1.6.3-1.amzn2023.0.2.x86_64.rpm 2.5 MB/s | 97 kB 00:00
(3/13): apr-util-lmdb-1.6.3-1.amzn2023.0.2.x86_64.rpm 327 kB/s | 13 kB 00:00
=====

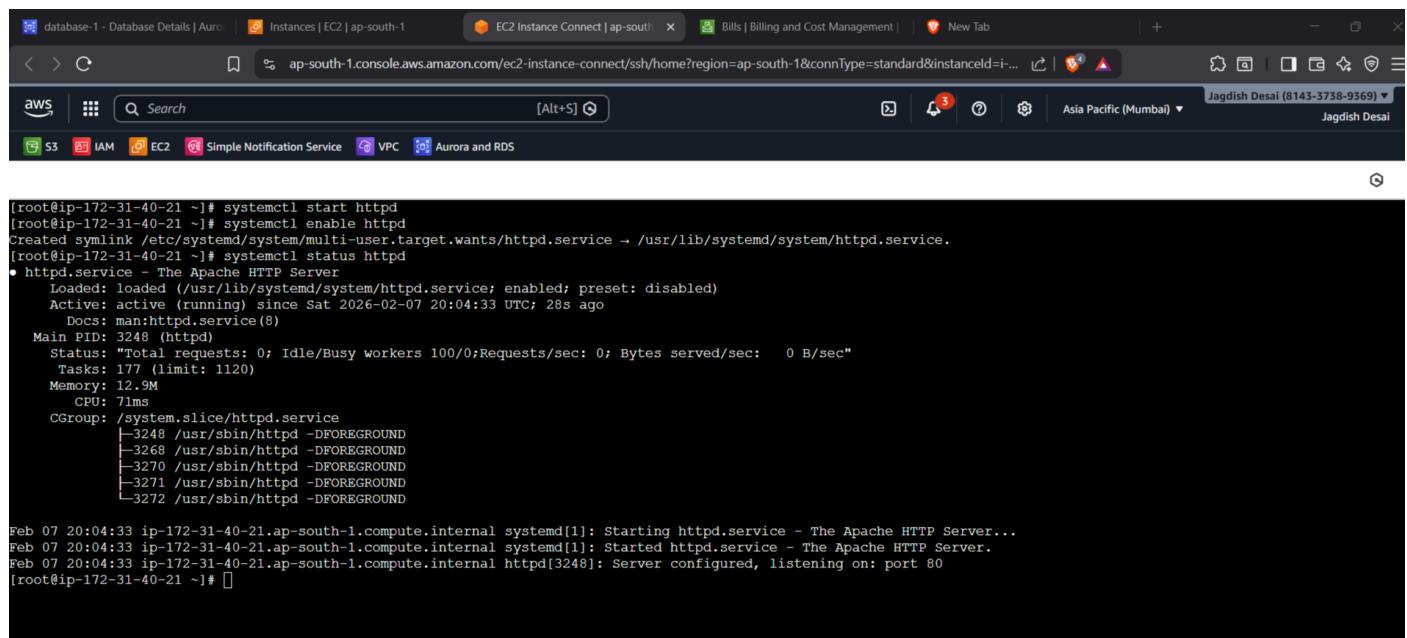
```

## Starting httpd:

systemctl start httpd

systemctl enable httpd

systemctl status httpd



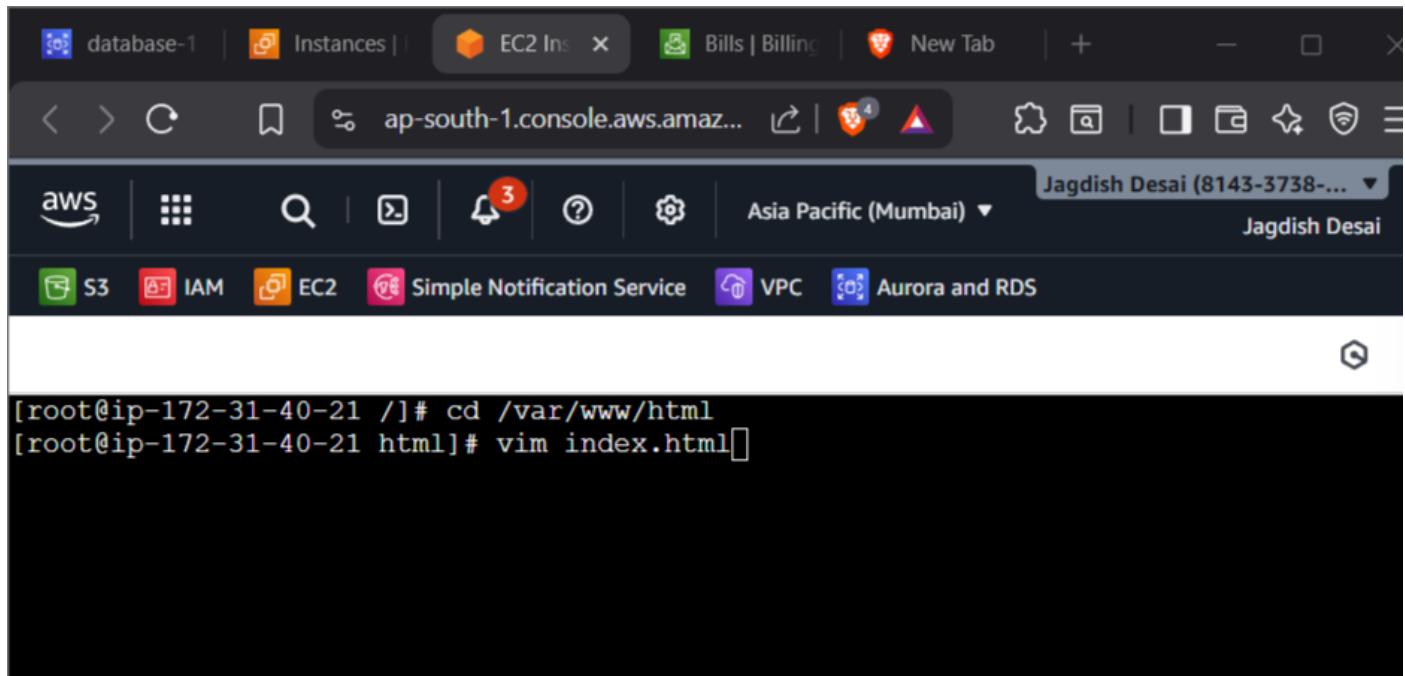
```
[root@ip-172-31-40-21 ~]# systemctl start httpd
[root@ip-172-31-40-21 ~]# systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[root@ip-172-31-40-21 ~]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
   Active: active (running) since Sat 2026-02-07 20:04:33 UTC; 28s ago
     Docs: man:httpd.service(8)
 Main PID: 3248 (httpd)
   Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec: 0 B/sec"
    Tasks: 177 (limit: 1120)
   Memory: 12.9M
      CPU: 71ms
   CGroup: /system.slice/httpd.service
           ├─3248 /usr/sbin/httpd -DFOREGROUND
           ├─3268 /usr/sbin/httpd -DFOREGROUND
           ├─3270 /usr/sbin/httpd -DFOREGROUND
           ├─3271 /usr/sbin/httpd -DFOREGROUND
           └─3272 /usr/sbin/httpd -DFOREGROUND

Feb 07 20:04:33 ip-172-31-40-21.ap-south-1.compute.internal systemd[1]: Starting httpd.service - The Apache HTTP Server...
Feb 07 20:04:33 ip-172-31-40-21.ap-south-1.compute.internal systemd[1]: Started httpd.service - The Apache HTTP Server.
Feb 07 20:04:33 ip-172-31-40-21.ap-south-1.compute.internal httpd[3248]: Server configured, listening on: port 80
[root@ip-172-31-40-21 ~]# 
```

**Step 3: After installation of httpd go to to default website folder in linux.**

```
cd /var/www/html
```

```
vim index.html
```



```
[root@ip-172-31-40-21 ~]# cd /var/www/html
[root@ip-172-31-40-21 html]# vim index.html
```

**Inside That Index.html file we type our code that should displays our public ip address.**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Public IP Viewer</title>
    <style>
        body {
            margin: 0;
            height: 100vh;
            font-family: 'Segoe UI', Tahoma, sans-serif;
            display: flex;
            justify-content: center;
            align-items: center;
        }
    </style>

```

```
background: linear-gradient(135deg, #667eea, #764ba2);
color: white;
}

.card {
background: rgba(255, 255, 255, 0.15);
padding: 40px 60px;
border-radius: 15px;
text-align: center;
box-shadow: 0 10px 25px rgba(0,0,0,0.3);
backdrop-filter: blur(8px);
}

h1 {
margin-bottom: 15px;
font-size: 32px;
}

.ip {
font-size: 22px;
font-weight: bold;
color: #ffeb3b;
margin-top: 10px;
}

footer {
margin-top: 20px;
font-size: 14px;
opacity: 0.8;
}

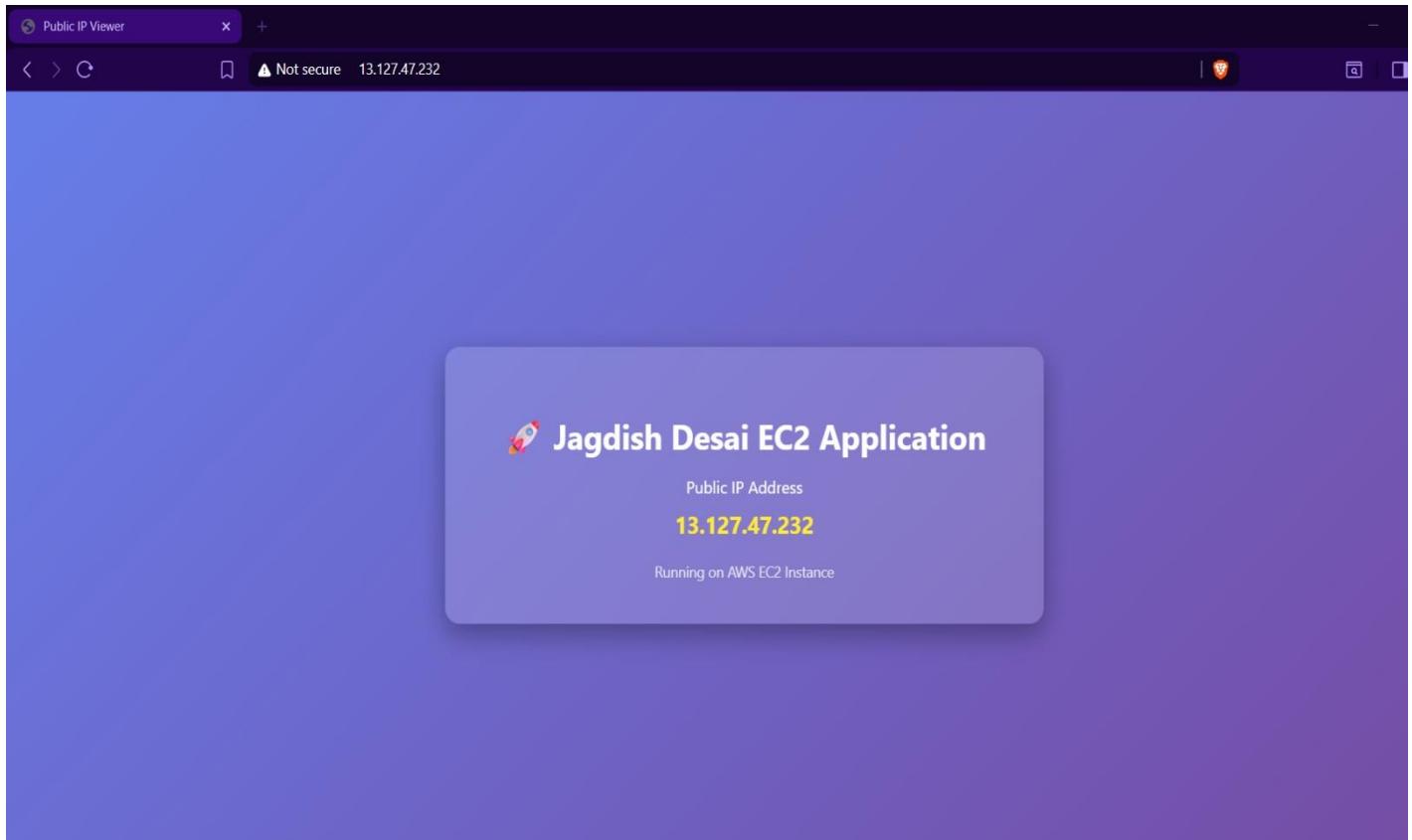
</style>
</head>
<body>
<div class="card">
```

```
<h1>🚀 Jagdish Desai EC2 Application</h1>
<div>Public IP Address</div>
<div class="ip" id="publicIp">Loading...</div>
<footer>Running on AWS EC2 Instance</footer>
</div>
<script>
  document.getElementById("publicIp").innerText =
    window.location.hostname || "Localhost";
</script>
</body>
</html>
```

**Step 4: Run application by using public ip of our instance and paste it on browser.**

**http://13.127.47.232/**

**you can see ip of our instance and application[same]:**



# Accessing Application Via Elastic Ip:

## Step 1: Go To Elastic IP's

The screenshot shows the AWS Management Console interface for the EC2 service. The left sidebar is expanded, showing categories like 'Elastic Block Store', 'Network & Security' (with 'Elastic IPs' selected), 'Load Balancing', and 'Auto Scaling'. The main content area is titled 'Elastic IP addresses' and displays a search bar and filter options for 'Name', 'Allocated IPv4 addr...', 'Type', and 'Allocation ID'. A message at the top right says 'No Elastic IP addresses found in this Region'. Below this, a section titled 'Select an elastic IP address' contains a note about viewing IP address usage with 'Public IP insights'.

## Step 2: Clicking On allocate elastic ip

The screenshot shows the 'Allocate Elastic IP address' configuration page. It includes sections for 'Elastic IP address settings' (with options for 'Amazon's pool of IPv4 addresses', 'Public IPv4 address', 'Customer-owned pool', and 'Allocate using an IPv4 IPAM pool'), 'Network border group' (set to 'ap-south-1'), 'Global static IP addresses' (with a link to 'AWS Global Accelerator'), and 'Tags - optional' (with a note about tags and an 'Add new tag' button). The bottom of the page shows standard navigation links like CloudShell, Feedback, and Console Mobile App, along with copyright information for Amazon Web Services.

## Step 3: Click on associate elastic ip address.

The screenshot shows the AWS EC2 console with the 'Elastic IP addresses' page. A green success message box at the top left says 'Elastic IP address allocated successfully. Elastic IP address 13.232.14.227'. To the right of the message is a button labeled 'Associate this Elastic IP address'. On the far right, a context menu is open under the 'Actions' heading, with 'Accept transfers' highlighted. The main table lists one elastic IP address: 13.232.14.227, which is a Public IP with allocation ID eipalloc-002885ca32407288d. The 'Summary' tab is selected. The left sidebar shows navigation links for AMI Catalog, Elastic Block Store, Network & Security (selected), Load Balancing, Auto Scaling, and Settings.

The screenshot shows the 'Associate Elastic IP address' dialog box. At the top, it displays the elastic IP address: 13.232.14.227. Below that, there's a section for 'Resource type' with two options: 'Instance' (selected) and 'Network interface'. A warning message states: 'If you associate an Elastic IP address with an instance that already has an Elastic IP address associated, the previously associated Elastic IP address will be disassociated, but the address will still be allocated to your account.' A note below says: 'If no private IP address is specified, the Elastic IP address will be associated with the primary private IP address.' The 'Instance' field contains the ID i-02855edd599308d61. The 'Private IP address' field contains 172.31.9.81. Under 'Reassociation', there's a checkbox for 'Allow this Elastic IP address to be reassociated'. At the bottom right are 'Cancel' and 'Associate' buttons.

The screenshot shows the AWS EC2 console with the 'Elastic IP addresses' section selected. A green success message at the top states: 'Elastic IP address associated successfully. Elastic IP address 13.232.14.227 has been associated with instance i-02855edd599308d61'. Below this, a table lists one elastic IP address: 'Public IPv4 address : 13.232.14.227' (Allocated IPv4 address), 'Type : Public IP', 'Allocation ID : eipalloc-002885ca32407288d', and 'Reverse DNS record : -'. There is also a link to 'Actions' and 'Allocate Elastic IP address'.

**Step 4 : See That New Ip Address Will Be allocated that will not change after restarting the instance.**

**Old ip- 13.127.47.232**

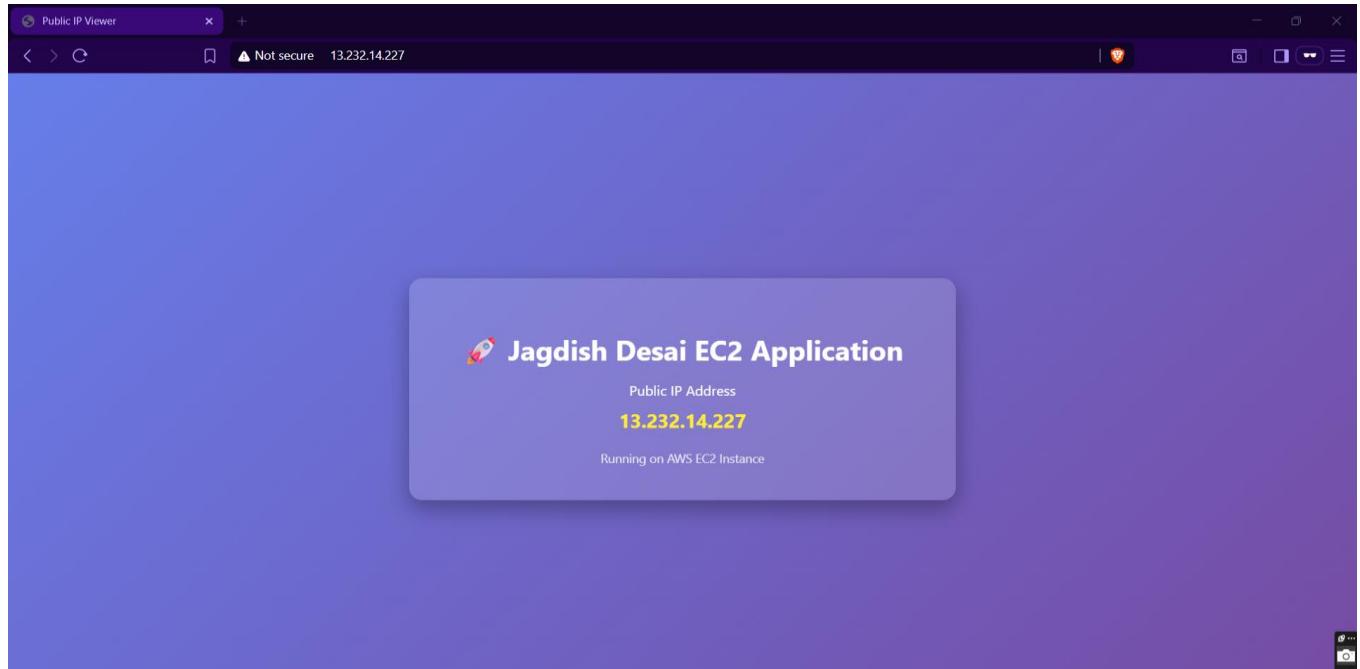
**New ip- 13.232.14.227**

The screenshot shows the AWS EC2 console with the 'Instances' section selected. It displays a single instance named 'task-4' with the following details: Instance ID: i-0fe3ecdfbdcc54d7a, Instance state: Running, Instance type: t2.micro, Status check: Initializing, Alarm status: View alarms +, Availability Zone: ap-south-1a, and Public IP: ec2-13-232-14-227.ap-south-1.compute.amazonaws.com. The 'Details' tab is selected in the instance summary.

## Step 5 : Using That New Ip We are accesing our application

Paste New ip- 13.232.14.227 on browser.

**http://13.232.14.227/**



You Will See in the output Public ip address **13.232.14.227**