

Creating Load Balancer

1. Search Ec2 in aws search bar

We can select different zone(mumbai).

2. Create 2 Instances server1 and server2.

Number of instances: 1

Software Image (AMI): Amazon Linux 2023 AMI 2023.4.2...
ami-0c101f26f147fa7fd

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

Launch instance

3. Select Default AMI Amazon Linux

Number of instances: 1

Software Image (AMI): Amazon Linux 2023 AMI 2023.4.2...
ami-0c101f26f147fa7fd

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

Launch instance

4. Default values select don't touch/ changes

The screenshot shows the AWS Launch Wizard interface. In the top left, the 'Amazon Machine Image (AMI)' section shows 'Amazon Linux 2023 AMI' with the ID 'ami-0c101f26f147fa7fd'. Below it, the 'Instance type' section shows 't2.micro' selected. On the right, a summary panel indicates 'Number of instances: 1' and shows the 'Free tier: In your first year' offer. A prominent orange 'Launch instance' button is at the bottom right.

5. Select your new key pair and store your folder where you running .bashrc

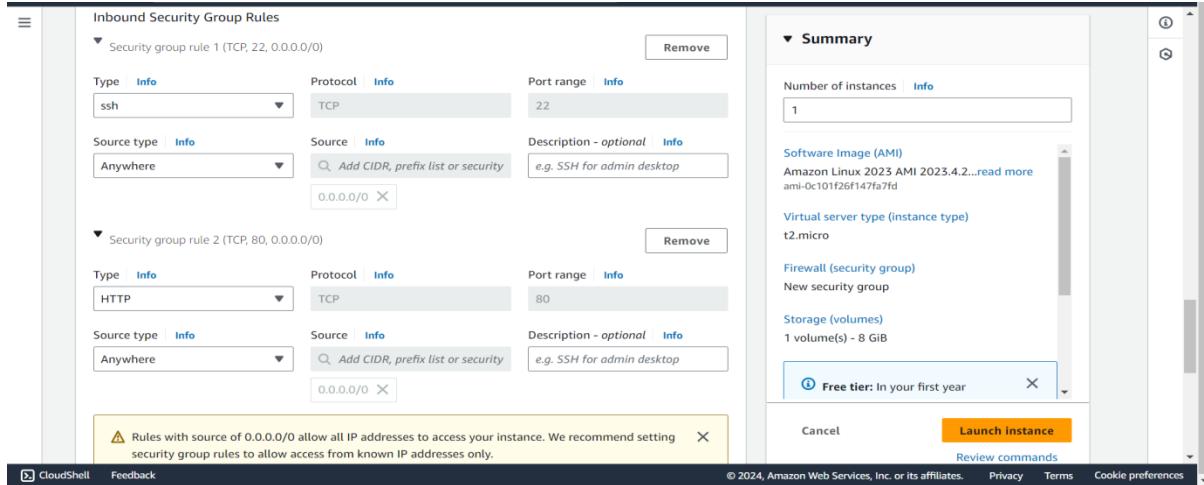
This screenshot continues the AWS Launch Wizard. It shows the 'Key pair (login)' step with 'Server1&2' selected. The 'Network settings' step follows, showing the VPC 'vpc-0a491cb5a5a3f1384' and subnet 'No preference'. The summary panel on the right remains consistent with one instance and the free tier offer.

6. Edit network settings(Default)

The final screenshot shows the 'Network settings' step in detail. It includes fields for VPC ('vpc-0a491cb5a5a3f1384'), subnet ('No preference'), and auto-assign public IP ('Enable'). It also provides options to 'Create security group' or 'Select existing security group', with a field for the name 'launch-wizard-14'. The summary panel on the right shows the same configuration with the 'Free tier: In your first year' offer.

7. Add Security group default leave it, add new one Type field select HTTP and source type Anywhere.

8. Click on Launch instance



9. Connect through ssh server1 in local [bashrc](#).

10. [Installing httpd server1](#)

The screenshot displays two terminal sessions. The left session shows the installation of the httpd package and its dependencies:

```
DELL_LAPTOP9DESKTOP-2035130 MINGW64 ~/Desktop/New folder (new_branch)
$ ssh -i "server1.pem" ec2-user@ec2-3-110-168-104.ap-south-1.compute.amazonaws.com
The authenticity of host 'ec2-3-110-168-104.ap-south-1.compute.amazonaws.com (64.179.206.36)' can't be established.
ED25519 key fingerprint is SHA256:M+Dg297Aljnrl6fLRLPT/jhlc+kIA/z1wAGUSTZnfPwp.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-110-168-104.ap-south-1.compute.amazonaws.com' (ED25519) to the list of known hosts.
[...]
[ec2-user@ip-172-31-15-246 ~]$ sudo su
[root@ip-172-31-15-246 ec2-user]# yum install httpd -y
Last metadata expiration check: 0:01:59 ago on Tue Apr 2 13:35:53 2024.
Dependencies resolved.
[...]
```

The right session shows the transaction summary for installing 12 packages, including httpd and its dependencies like apr, apr-util, generic-logos-httpd, etc. The transaction summary table includes columns for Package, Arch, Version, Repository, and Size.

Package	Arch	Version	Repository	Size
httpd	x86_64	2.4.58-1.amzn2023	amazonlinux	47 k
apr	x86_64	1.7.2-2.amzn2023.0.2	amazonlinux	129 k
apr-util	x86_64	1.6.3-1.amzn2023.0.1	amazonlinux	98 k
generic-logos-httpd	noarch	18.0.0-1.amzn2023.0.3	amazonlinux	19 k
httpd-core	x86_64	2.4.58-1.amzn2023	amazonlinux	1.4 M
httpd-filesystem	noarch	2.0.1-1.amzn2023	amazonlinux	14 k
httpd-tools	x86_64	2.4.58-1.amzn2023	amazonlinux	81 k
libbrotli	x86_64	1.0.9-4.amzn2023.0.2	amazonlinux	315 k
mod_lua	noarch	2.1.49-3.amzn2023.0.3	amazonlinux	33 k
[...]				
apr-util-openssl	x86_64	1.6.3-1.amzn2023.0.1	amazonlinux	17 k
mod_http2	x86_64	2.0.11-2.amzn2023	amazonlinux	150 k
mod_lua	x86_64	2.4.58-1.amzn2023	amazonlinux	61 k

11. Check httpd services after installed

The screenshot shows two terminal sessions. The left session shows the status of the httpd service and its dependencies:

```
[root@ip-172-31-15-246 ~]$ sudo systemctl status httpd
[...]
httpd.service - The Apache HTTP Service
  Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: off)
  Active: active (running) since Tue 2024-04-02 13:38:20 UTC; 31s ago
    Docs: man:httpd.service(8)
  Main PID: 25557 (httpd)
    Tasks: 177 (limit: 1114)
      Memory: 800KiB
        CPU: 0.000 CPU(s)
      CGroup: /system.slice/httpd.service
              └─25558 /usr/sbin/httpd -DFOREGROUND
[...]
```

The right session shows the log output for the httpd service, which indicates it has started successfully and is listening on port 80.

```
Apr 02 13:38:20 ip-172-31-15-246.ap-south-1.compute.internal systemd[1]: Started The Apache HTTP Service.
Apr 02 13:38:20 ip-172-31-15-246.ap-south-1.compute.internal systemd[1]: Started httpd[25557].
[END]
```

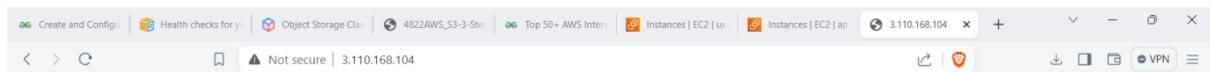
```

[root@ip-172-31-15-246 ec2-user]# service httpd status
Redirecting to /bin/systemctl status httpd.service
● httpd.service - The Apache HTTP Server
    Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
      Active: active (running) since Tue 2024-04-02 13:38:20 UTC; 1min 29s ago
        Docs: man:httpd.service(8)
    Main PID: 25557 (httpd)
      Status: "total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec: 0"
         Tasks: 177 (limit: 1114)
        Memory: 13.1M
          CPU: 114ms
        CGroup: /system.slice/httpd.service
            ├─25557 /usr/sbin/httpd -DFOREGROUND
            ├─25558 /usr/sbin/httpd -DFOREGROUND
            ├─25559 /usr/sbin/httpd -DFOREGROUND
            ├─25560 /usr/sbin/httpd -DFOREGROUND
            └─25561 /usr/sbin/httpd -DFOREGROUND

Apr 02 13:38:20 ip-172-31-15-246.ap-south-1.compute.internal systemd[1]: Starting httpd.service
Apr 02 13:38:20 ip-172-31-15-246.ap-south-1.compute.internal systemd[1]: Started httpd.service
Apr 02 13:38:20 ip-172-31-15-246.ap-south-1.compute.internal httpd[25557]: Server configured
lines 1-19/19 (END)
AC
[root@ip-172-31-15-246 ec2-user]# cd /var/www/html
[root@ip-172-31-15-246 html]# vi index.html
[root@ip-172-31-15-246 html]# service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[root@ip-172-31-15-246 html]#

```

12. Using public ip address to run on webpage



Welcome to webserver1

1. Create same as it Server1. Create server2.

EC2 > instances > Launch an instance

Name and tags

Name: server2

Launch an instance

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.

Application and OS Images (Amazon Machine Image)

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.

Recents

Quick Start

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI
ami-0c101f26f147fa7fd (64-bit (x86), uefi-preferred) / ami-0d8f91fa8ecd3b58 (64-bit (Arm), uefi)
Virtualization: hvm ENA enabled: true Root device type: ebs
Free tier eligible

Services

Description

Amazon Linux 2023 AMI 2023.4.20240319.1 x86_64 HVM kernel-6.1

Architecture	Boot mode	AMI ID
64-bit (x86)	uefi-preferred	ami-0c101f26f147fa7fd

Key pair (login)

Server1&2

Network settings

VPC - required
vpc-0a491cb5a5af1384 (default)
Subnet
No preference
Auto-assign public IP
Enable
Additional charges apply when outside of free tier allowance
Firewall (security groups)

The left pane shows the 'Network settings' configuration for a VPC. It includes fields for VPC ID (vpc-0a491cb5a3f1384), Subnet ID (172.31.0.0/16), Auto-assign public IP (Enabled), and Firewall (security groups) (Create security group). A note states: 'Additional charges apply when outside of free tier allowance'. Below this is a section for 'Security group name - required' containing 'launch-wizard-15'. A note says: 'This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and _/-@#{}&{}/<{>}'.

The right pane shows the 'Inbound Security Group Rules' configuration for a security group rule 1 (TCP 22, 0.0.0.0/0). It has two entries: one for TCP port 22 from 'Anywhere' and another for HTTP port 80 from 'Anywhere'. Both have optional descriptions: 'e.g. SSH for admin desktop' and 'e.g. SSH for admin desktop' respectively. A note at the bottom says: 'Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting a specific IP or CIDR range.'

2. Launch server2 Instance

The left pane shows the 'Configure storage' section of the instance creation wizard. It specifies 1x 8 GiB gp3 volume as the root volume (Not encrypted). A note indicates: 'Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage'. Below this is an 'Advanced details' section.

The right pane shows the 'Summary' section of the instance creation wizard. It lists 'Number of instances' (1), 'Software Image (AMI)' (Amazon Linux 2023 AMI 2023.4.2...), 'Virtual server type (instance type)' (t2.micro), 'Firewall (security group)' (New security group), 'Storage (volumes)' (1 volume(s) - 8 GiB), and a note: 'Free tier: In your first year'. At the bottom are 'Cancel' and 'Launch Instance' buttons.

The bottom navigation bar includes CloudShell, Feedback, and links to Privacy, Terms, and cookie preferences.

The main navigation bar shows the AWS logo, Services, Search, and Mumbai region.

3. As it is httpd installed in server2, check status, restart, start httpd.

MINCw64: Shows a terminal session where the user runs 'sudo su' and then 'yum install httpd'. The output shows the package is already installed. The user then runs 'httpd -v' to verify the version.

DELL LAPTOP: Shows a terminal session where the user runs 'sudo su' and then 'yum install httpd'. The output shows the package is being downloaded and installed. The user then runs 'httpd -v' to verify the version.

```

MINGW64/c/Users/DELL LAPTOP/Desktop/New folder
Verifying : httpd-tools-2.4.58-1.amzn2023.x86_64 1/12
Verifying : mod_lua-2.4.58-1.amzn2023.x86_64 2/12
Verifying : apr-1.7.2-2.amzn2023.0.2.x86_64 3/12
Verifying : httpd-2.4.58-1.amzn2023.x86_64 4/12
Verifying : apr-util-1.6.3-1.amzn2023.0.1.x86_64 5/12
Verifying : mod_http2-2.0.11-2.amzn2023.x86_64 6/12
Verifying : apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64 7/12
Verifying : httpd-core-2.4.58-1.amzn2023.x86_64 8/12
Verifying : libbrotli-1.0.9-1.amzn2023.0.2.x86_64 9/12
Verifying : mailcap-2.1.49-3.amzn2023.0.3.noarch 10/12
Verifying : generic-logos-httdp-18.0.0-12.amzn2023.0.3.noarch 11/12
Verifying : httpd-filesystem-2.4.58-1.amzn2023.noarch 12/12

Installed:
apr-1.7.2-2.amzn2023.0.2.x86_64
apr-util-1.6.3-1.amzn2023.0.1.x86_64
apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64
generic-logos-httdp-18.0.0-12.amzn2023.0.3.noarch
httpd-2.4.58-1.amzn2023.x86_64
httpd-core-2.4.58-1.amzn2023.x86_64
httpd-filesystem-2.4.58-1.amzn2023.noarch
httpd-tools-2.4.58-1.amzn2023.x86_64
libbrotli-1.0.9-1.amzn2023.0.2.x86_64
mailcap-2.1.49-3.amzn2023.0.3.noarch
mod_http2-2.0.11-2.amzn2023.x86_64
mod_lua-2.4.58-1.amzn2023.x86_64

Complete!
[root@ip-172-31-9-121 ec2-user]# service httpd start
Redirecting to /bin/systemctl start httpd.service
[root@ip-172-31-9-121 ec2-user]# service httpd status
Redirecting to /bin/systemctl status httpd.service
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: d
     Active: active (running) since Tue 2024-04-02 13:45:49 UTC; 10s ago
       Docs: man:httpd.service(8)
   Main PID: 25587 (httpd)
     Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes
   Tasks: 177 (limit: 1114)
   Memory: 13.1M
      CPU: 67ms
      CGroup: /system.slice/httpd.service
           ├─25587 /usr/sbin/httpd -DFOREGROUND
           ├─25588 /usr/sbin/httpd -DFOREGROUND
           ├─25589 /usr/sbin/httpd -DFOREGROUND
           ├─25590 /usr/sbin/httpd -DFOREGROUND
           ├─25591 /usr/sbin/httpd -DFOREGROUND

```



```

MINGW64/c/Users/DELL LAPTOP/Desktop/New folder
^C
[root@ip-172-31-9-121 ec2-user]# service httpd status
Redirecting to /bin/systemctl status httpd.service
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
     Active: active (running) since Tue 2024-04-02 13:45:49 UTC; 59s ago
       Docs: man:httpd.service(8)
   Main PID: 25587 (httpd)
     Status: "total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec: 0 B/sec"
   Tasks: 177 (limit: 1114)
   Memory: 13.1M
      CPU: 97ms
      CGroup: /system.slice/httpd.service
           ├─25587 /usr/sbin/httpd -DFOREGROUND
           ├─25588 /usr/sbin/httpd -DFOREGROUND
           ├─25589 /usr/sbin/httpd -DFOREGROUND
           ├─25590 /usr/sbin/httpd -DFOREGROUND
           ├─25591 /usr/sbin/httpd -DFOREGROUND

Apr 02 13:45:49 ip-172-31-9-121.ap-south-1.compute.internal systemd[1]: Starting httpd.service - The Apache HTTP Server...
Apr 02 13:45:49 ip-172-31-9-121.ap-south-1.compute.internal systemd[1]: Started httpd.service - The Apache HTTP Server.
Apr 02 13:45:49 ip-172-31-9-121.ap-south-1.compute.internal httpd[25587]: Server configured, listening on: port 80
[root@ip-172-31-9-121 ec2-user]# cd /var/www/html
[root@ip-172-31-9-121 html]# vi index.html

```

4. Select server2 public ip address go to webpage paste click on enter



Welcome server2

1. Creating loadbalancer for server1 and server2 instances

1. First we creating Target group for servers1&2, click on create target group.

The screenshot shows the AWS EC2 Target groups page. On the left, there's a navigation sidebar with categories like AMI Catalog, Elastic Block Store, Network & Security, Load Balancing, Auto Scaling, and Target Groups (which is currently selected). The main area is titled "Target groups" and shows a table with columns for Name, ARN, Port, Protocol, and Target type. A message at the bottom says "No target groups" and "You don't have any target groups in ap-south-1". At the top right, there's a "Create target group" button. The URL in the address bar is <https://ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#TargetGroups>.

2. Select target type, hear Instances selected

The screenshot shows the "Specify group details" step of the "Create target group" wizard. It has two tabs: "Step 1 Specify group details" (selected) and "Step 2 Register targets". Under "Basic configuration", it says "Your load balancer routes requests to the targets in a target group and performs health checks on the targets." In the "Choose a target type" section, the "Instances" option is selected, which includes a note about supporting EC2 Auto Scaling. Other options like "IP addresses" and "Lambda function" are also listed. The URL in the address bar is [## 3. Give Target group name](https://ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#TargetGroups>Create target group.</p></div><div data-bbox=)

The screenshot shows the final step of the "Create target group" wizard. It shows the "Target group name" field filled with "server1-2-tg", the "Protocol" dropdown set to "HTTP", and the "Port" dropdown set to "80". Below these, there's a note about IP address type and a note about IPv4. The URL in the address bar is Create target group.

4. Remaining are Default don't disturb other values

The screenshot shows the 'Protocol version' section with 'HTTP1' selected. It also shows the 'Health checks' section with 'Health check protocol' set to 'HTTP'.

5. Click on next

The screenshot shows the 'Health check path' field set to '/' and the 'Attributes' section with a note about default attributes.

6. Register targets select both created instance ID, click on Include as pending below

The screenshot shows the 'Available instances' table with two instances selected: 'server2' and 'server1'. Below the table, the message '2 selected' is displayed, followed by a port selection field containing '80' and the instruction '1-65535 (separate multiple ports with commas)'. At the bottom is the button 'Include as pending below'.

7. Click on Create target group

The screenshot shows the AWS CloudFront console with the 'Create target group' wizard. In the 'Review targets' step, two targets are listed: 'server2' and 'server1'. Both are running instances with port 80. The 'Create target group' button is highlighted at the bottom right.

Instance ID	Name	Port	State	Security groups	Zone	Private IPv4 address
i-066e667ca50ce8620	server2	80	Running	launch-wizard-2	ap-south-1b	172.31.9.121
i-01f49cc92dfc3ed55	server1	80	Running	launch-wizard-1	ap-south-1b	172.31.15.246

8. Successfully created our target group

The screenshot shows the AWS EC2 Target Groups console. A success message indicates the target group 'server1-2-tg' was created successfully. The 'Details' section shows the target type as 'Instance', protocol as 'HTTP: 80', and load balancer as 'None associated'. The VPC is listed as 'vpc-09bf1ce5a6bccd291'.

Target type	Protocol : Port	Protocol version	VPC
Instance	HTTP: 80	HTTP1	vpc-09bf1ce5a6bccd291

1. Now we Creating Load balancer.

1. Go to left navigation bar scroll down select Load Balancers, click on Create load balancer.

Introducing resource map for Application Load Balancers

Resource map is a visual representation of the relationships between load balancer resources and provides the ability to view, explore, and troubleshoot the architecture of your load balancer. Resource map can be viewed on the load balancers detail page. Share feedback to help us improve your experience.

Load balancers

Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

Filter load balancers

Name	DNS name	State	VPC ID	Availability Zones
No load balancers				

You don't have any load balancers in ap-south-1

Create load balancer

0 load balancers selected

Select a load balancer above.

2. Select load balancer type Application Load Balancer, click on create link button

Compare and select load balancer type

A complete feature-by-feature comparison along with detailed highlights is also available. [Learn more](#)

Load balancer types

Application Load Balancer	Network Load Balancer	Gateway Load Balancer

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

3. Give Load balancer name

Create Application Load Balancer

The Application Load Balancer distributes incoming HTTP and HTTPS traffic across multiple targets such as Amazon EC2 instances, microservices, and containers, based on request attributes. When the load balancer receives a connection request, it evaluates the listener rules in priority order to determine which rule to apply, and if applicable, it selects a target from the target group for the rule action.

How Application Load Balancers work

Basic configuration

Load balancer name

Name must be unique within your AWS account and can't be changed after the load balancer is created.

app_loadbalancer

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Scheme [Info](#)

Scheme can't be changed after the load balancer is created.

Internet-facing

An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. [Learn more](#)

Internal

An internal load balancer routes requests from clients to targets using private IP addresses.

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Default values don't disturb

The screenshot shows the AWS Load Balancer configuration interface. Under 'IP address type', 'IPv4' is selected. In the 'Network mapping' section, a single VPC entry is shown: 'vpc-09bf1ce5a6bcdd291' with 'IPv4 VPC CIDR: 172.31.0.0/16'. Under 'Mappings', three Availability Zones are listed: 'ap-south-1a (aps1-az1)', 'ap-south-1b (aps1-az3)', and 'ap-south-1c (aps1-az2)'. Each has a 'Subnet' dropdown set to 'subnet-0a7f340a279224c59'. The bottom of the screen shows standard AWS navigation links: CloudShell, Feedback, Privacy, Terms, and Cookie preferences.

4. Select Mappings, Availability Zones

This screenshot shows the same configuration interface as above, but with more mappings selected. In the 'Mappings' section, all three Availability Zones ('ap-south-1a', 'ap-south-1b', and 'ap-south-1c') are checked. Each has its corresponding 'Subnet' dropdown set to 'subnet-0a7f340a279224c59'. The bottom of the screen shows standard AWS navigation links: CloudShell, Feedback, Privacy, Terms, and Cookie preferences.

1. Create security group using the link in load balancer go through that give name.

The screenshot shows the 'Create security group' page in the AWS EC2 console. The 'Basic details' section includes fields for 'Security group name' (set to 'Allow Http'), 'Description' (set to 'Allows SSH access to developers'), and 'VPC' (set to 'vpc-09bf1ce5a6bcdd291'). A note at the top states: 'A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.' The bottom of the screen shows standard AWS navigation links: EC2 > Security Groups > Create security group, and CloudShell, Feedback, Privacy, Terms, and Cookie preferences.

2. Give group name, Description, default VPC , Add Inbound rules give HTTP, source anywhere ip4

The screenshot shows the 'Create security group' wizard. In the 'Basic details' step, a new security group is being created with the name 'Allow Http'. It has a description 'server1&2' and is associated with the VPC 'vpc-09bf1ce5a6bcdd291'. In the 'Inbound rules' step, a single rule is defined: Type: HTTP, Protocol: TCP, Port range: 80, Source: Any..., and Destination: 0.0.0.0/0. A 'Delete' button is visible next to the rule.

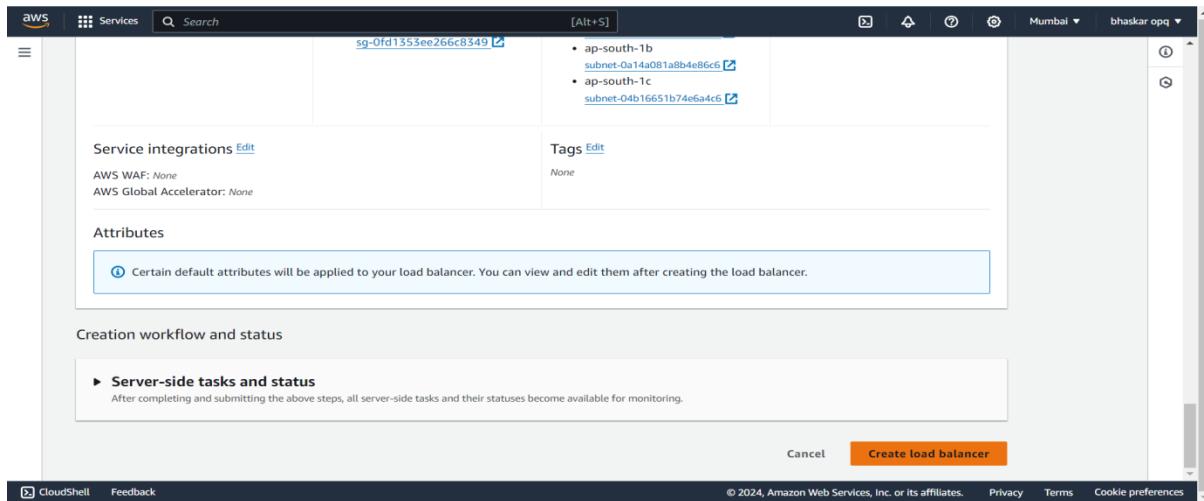
Click on Create security group

The screenshot shows the continuation of the 'Create security group' wizard. Under 'Tags - optional', there are no tags assigned. At the bottom right, the 'Create security group' button is highlighted in orange.

5. Again we came back load balancer page select security groups Allow Http our created

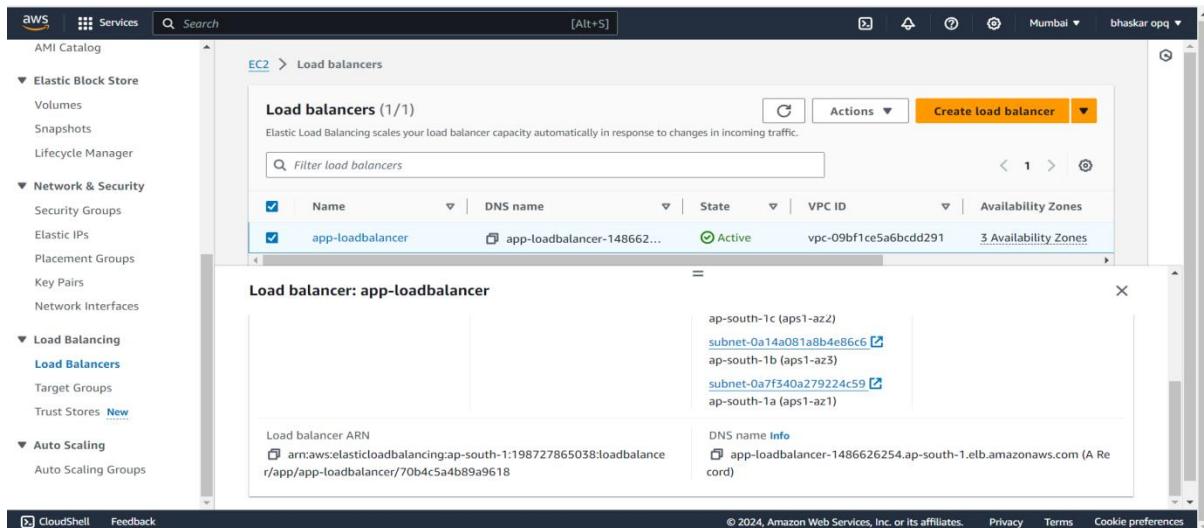
The screenshot shows the 'Listeners and routing' section of the AWS Load Balancers console. A listener named 'HTTP:80' is configured to forward traffic to a target group named 'server1-2-tg'. The target group is associated with the security group 'Allow Http'.

6. Click on Create Load balancer



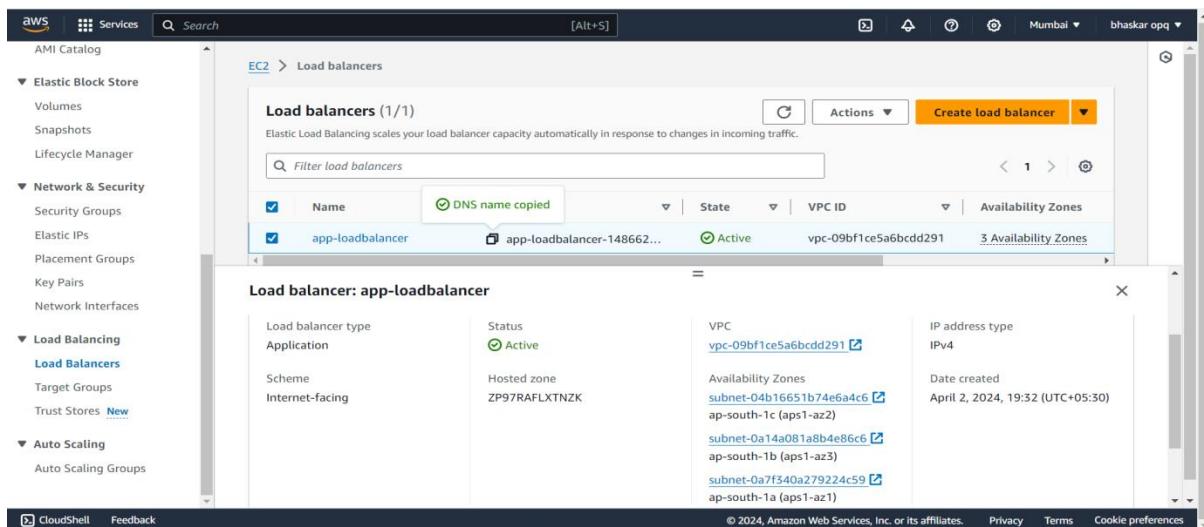
The screenshot shows the AWS CloudFormation console. A modal window titled 'Create stack' is open. In the 'Template' tab, there is a large JSON template. Below it, the 'Outputs' section contains several output keys with their corresponding values. At the bottom right of the modal, there is a large orange 'Create stack' button.

7. Load Balancer was created, state was Active(taking few minutes)



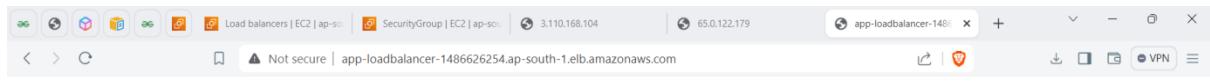
The screenshot shows the AWS EC2 Load Balancers page. On the left, a navigation menu includes 'Load Balancing' under 'Load Balancers'. The main table lists one load balancer: 'app-loadbalancer' (Status: Active, VPC ID: vpc-09bf1ce5a6bcd291, 3 Availability Zones). The 'Actions' dropdown for this load balancer has 'Edit' selected. A detailed view of the 'app-loadbalancer' is shown below the table, including its ARN and DNS name.

8. Copy DNS name



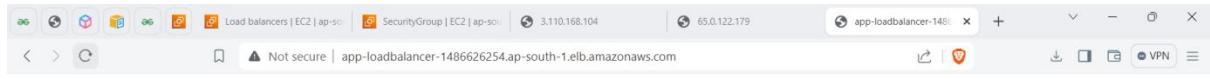
The screenshot shows the AWS EC2 Load Balancers page. The 'Actions' dropdown for the 'app-loadbalancer' load balancer has 'Copy DNS name' selected. A message 'DNS name copied' is displayed above the table. The table shows the same information as the previous screenshot. A detailed view of the 'app-loadbalancer' is shown below the table, including its ARN and DNS name. The 'Actions' dropdown for this load balancer has 'Edit' selected.

9. go to webpage Paste DNS URL we will get our created file in server1



Welcome to webserver1

10. Refresh the page then we will get server2 file data on the webpage



Welcome server2

Created data in the bashrc command

```
root@ip-172-31-15-246:/var/www/html
<h1>welcome to webserver1</h1>
```
exit
[ec2-user@ip-172-31-15-246 ~]$ exit
logout
Connection to ec2-3-110-168-104.ap-south-1.compute.amazonaws.com closed.

DELL_LAPTOP@DESKTOP-2035130 MINGW64 ~/Desktop/New folder (new_branch)
$ ssh -i "server1.pem" ec2-user@ec2-3-110-168-104.ap-south-1.compute.amazonaws.com
 _#
 \###_
  ~~~\###\
  ~~ \##|
  ~~  \#/  https://aws.amazon.com/linux/amazon-linux-2023
  ~~ \~. \_/
  ~~ / \_/
  ~~ /m'` 

Last login: Tue Apr  2 13:36:53 2024 from 157.45.242.131
[ec2-user@ip-172-31-15-246 ~]$ sudo su
[root@ip-172-31-15-246 ec2-user]# ls
[root@ip-172-31-15-246 ec2-user]# cd /var/www/html
[root@ip-172-31-15-246 html]# ls
index.html
[root@ip-172-31-15-246 html]# client_loop: send disconnect: Connection reset by peer

DELL_LAPTOP@DESKTOP-2035130 MINGW64 ~/Desktop/New folder (new_branch)
$ ssh -i "server1.pem" ec2-user@ec2-3-110-168-104.ap-south-1.compute.amazonaws.com
  _#
  \###_
  ~~~\###\
 ~~ \##|
 ~~ \#/ https://aws.amazon.com/linux/amazon-linux-2023
 ~~ \~. _/
 ~~ / _/
 ~~ /m'`

Last login: Tue Apr 2 13:50:33 2024 from 157.45.242.131
[ec2-user@ip-172-31-15-246 ~]$ sudo su
[root@ip-172-31-15-246 ec2-user]# cd /var/www/html
[root@ip-172-31-15-246 html]# ls
index.html
[root@ip-172-31-15-246 html]# vi index.html
[root@ip-172-31-15-246 html]# |
```

```
root@ip-172-31-9-121:/var/www/html
<h1>welcome server2</h1>
```
-- INSERT --
```