

TASK – 19

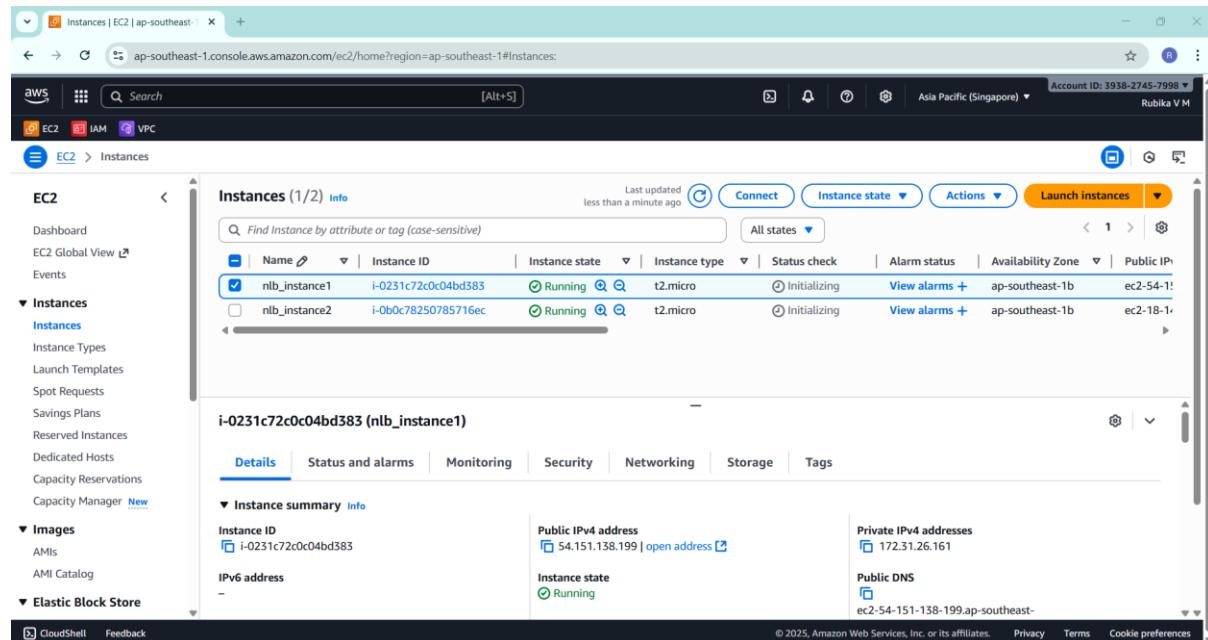
Elastic load balancer

Elastic Load Balancing (ELB) in AWS is a service that automatically distributes incoming application traffic across multiple targets, such as EC2 instances, containers, and IP addresses, within one or more Availability Zones. It enhances application availability and fault tolerance by ensuring that no single target is overwhelmed with traffic and by routing requests only to healthy targets.

Network load balancer

An AWS Network Load Balancer (NLB) is a high-performance load balancer that operates at the transport layer (OSI Layer 4) to distribute TCP, UDP, and TLS traffic across targets like EC2 instances and containers. It can handle millions of requests per second with ultra-low latency, making it ideal for high-throughput, network-intensive applications like gaming, streaming.

Creation of instance



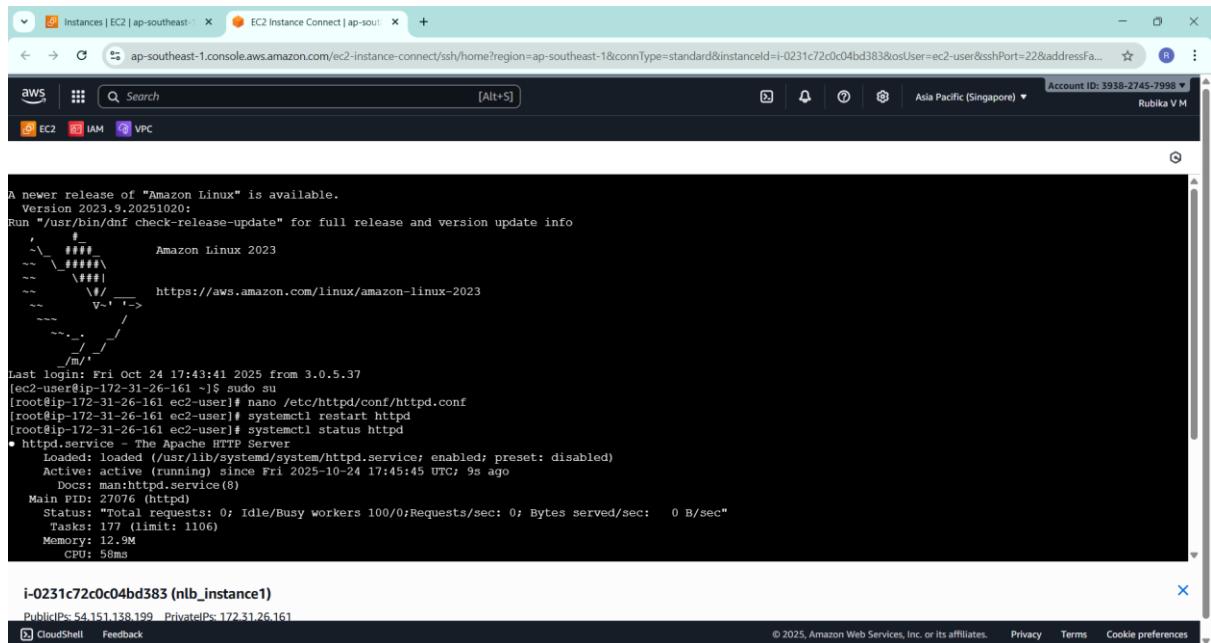
The screenshot shows the AWS EC2 Instances page. The left sidebar is collapsed, showing the EC2 navigation bar. The main content area displays a table of instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
nlb_instance1	i-0231c72c0c04bd383	Running	t2.micro	Initializing	View alarms +	ap-southeast-1b	ec2-54-1-
nlb_instance2	i-0b0c78250785716ec	Running	t2.micro	Initializing	View alarms +	ap-southeast-1b	ec2-18-1-

Below the table, the details for the first instance (i-0231c72c0c04bd383) are shown. The 'Details' tab is selected, displaying the following information:

- Instance summary**: Instance ID: i-0231c72c0c04bd383, Public IPv4 address: 54.151.138.199, Instance state: Running.
- Public IPv4 addresses**: 54.151.138.199.
- Private IPv4 addresses**: 172.31.26.161.
- Public DNS**: ec2-54-151-138-199.ap-southeast-

Changing port number in instance 1



A newer release of "Amazon Linux" is available.
Version 2023.9.20251020:
Run "/usr/bin/dnf check-release-update" for full release and version update info
Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023
Last login: Fri Oct 24 17:43:41 2025 from 3.0.5.37
(ec2-user@ip-172-31-26-161 ~)\$ sudo su
(root@ip-172-31-26-161 ec2-user)\$ nano /etc/httpd/conf/httpd.conf
(root@ip-172-31-26-161 ec2-user)\$ systemctl restart httpd
(root@ip-172-31-26-161 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 Fri 2025-10-24 17:45:45 UTC; 9s ago
 Docs: man:httpd.service(8)
 Main PID: 27076 (httpd)
 Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec: 0 B/sec"
 Tasks: 177 (limit: 1106)
 Memory: 12.9M
 CPU: 56ms

i-0231c72c0c04bd383 (nlb_instance1)
PublicIPs: 54.151.138.199 PrivateIPs: 172.31.26.161
CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



Changing port number in instance 2

Creation of Target group

The screenshot shows the AWS Management Console with the URL ap-southeast-1.console.aws.amazon.com/ec2/home?region=ap-southeast-1#TargetGroups. The page displays the 'Target groups' section under the 'Load Balancing' category in the left sidebar. There are two target groups listed:

Name	ARN	Port	Protocol	Target type	Load balancer	VPC ID
mytarget-2	arn:aws:elasticloadbalancing:ap-southeast-1:123456789012:targetgroup/mytarget-2/54321	85	HTTP	Instance	None associated	vpc-00a
mytarget-1	arn:aws:elasticloadbalancing:ap-southeast-1:123456789012:targetgroup/mytarget-1/54321	84	HTTP	Instance	None associated	vpc-00a

The left sidebar also includes sections for Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces, Load Balancers, Target Groups (which is selected), and Auto Scaling.

Creation of load balancer

The screenshot shows the AWS Cloud Console interface for creating a new load balancer. The left sidebar navigation includes 'EC2' under 'Load balancers'. The main content area displays a table titled 'Load balancers (1/1)' with one entry: 'my-nlb' (Active, network type, Internet-facing, IPv4, VPC vpc-00aa32b792ae81dee, 3 Availability Zones). Below this, the 'Load balancer: my-nlb' details page is shown with tabs for 'Details', 'Listeners', 'Network mapping', 'Resource map', 'Security', 'Monitoring', 'Integrations', 'Attributes', 'Capacity', and 'Tags'. The 'Details' tab shows the load balancer type (Network), status (Active), scheme (Internet-facing), VPC (vpc-00aa32b792ae81dee), availability zones (subnet-00b1afc1e5e29345, ap-), and date created (October 24, 2025, 23:38 (UTC+05:30)).

The screenshot shows the 'Edit inbound rules' page for a security group. It lists two existing rules: one for port 85 (Custom TCP, TCP, 85, Custom, 0.0.0.0/0) and another for port 84 (Custom TCP, TCP, 84, Custom, 0.0.0.0/0). A 'Delete' button is next to each rule. At the bottom, there is an 'Add rule' button, and 'Cancel', 'Preview changes', and 'Save rules' buttons.

Check for outcome through DNS



This is my NLB_1



This is my NLB_2

Hosting Nginx:

Creation of instance

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links like Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Capacity Manager, Images, AMIs, AMI Catalog, and Elastic Block Store. The main area has a title 'Instances (1/3) Info' with a search bar and filters for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IP. A table lists one instance: 'my_nginx' (Instance ID: i-0df5e7ab6da492ad0, State: Running, Type: t2.micro, Status: Initializing). Below the table is a detailed view for 'i-0df5e7ab6da492ad0 (my_nginx)' with tabs for Details, Status and alarms, Monitoring, Security, Networking, Storage, and Tags. Under 'Details', it shows Instance ID (i-0df5e7ab6da492ad0), Public IPv4 address (54.255.152.118), Private IPv4 addresses (172.31.24.75), and Instance state (Running). It also shows Public DNS (ec2-54-255-152-118.ap-southeast-1.compute.internal). At the bottom, there are links for CloudShell and Feedback.

Installing the package of Nginx

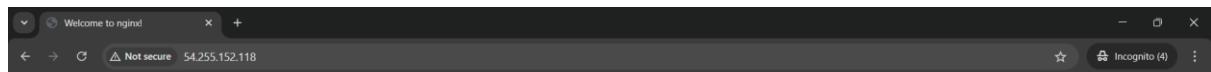
The screenshot shows the AWS EC2 Instance Connect terminal window. It displays a terminal session with the following command history and output:

```
Process: 26536 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
Process: 26537 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
Main PID: 26538 (nginx)
Tasks: 2 (limit: 1106)
Memory: 2.5M
CPU: 39ms
CGroup: /system.slice/nginx.service
└─26538 "nginx: master process /usr/sbin/nginx"
   └─26539 "nginx: worker process"

Oct 24 18:21:58 ip-172-31-24-75.ap-southeast-1.compute.internal systemd[1]: Starting nginx.service - The nginx HTTP and reverse proxy server...
Oct 24 18:21:58 ip-172-31-24-75.ap-southeast-1.compute.internal nginx[26536]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
Oct 24 18:21:58 ip-172-31-24-75.ap-southeast-1.compute.internal nginx[26536]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Oct 24 18:21:58 ip-172-31-24-75.ap-southeast-1.compute.internal systemd[1]: Started nginx.service - The nginx HTTP and reverse proxy server.
[root@ip-172-31-24-75 ec2-user]# systemctl restart nginx
[root@ip-172-31-24-75 ec2-user]# history
1  yum update -y
2  yum install nginx
3  yum install Ngnix
4  yum install Ngnix -y
5  yum update -y
6  yum install nginx -y
7  systemctl start nginx
8  systemctl status nginx
9  systemctl restart nginx
10 history
[root@ip-172-31-24-75 ec2-user]#
```

At the bottom of the terminal, it says 'i-0df5e7ab6da492ad0 (my_nginx)' and 'PublicIPs: 54.255.152.118 PrivateIPs: 172.31.24.75'. The footer includes links for CloudShell and Feedback.

Output



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](#).
Commercial support is available at [nginx.com](#).

Thank you for using nginx.