



ASSIGNMENT - 02

COURSE : AWS DEVOPS

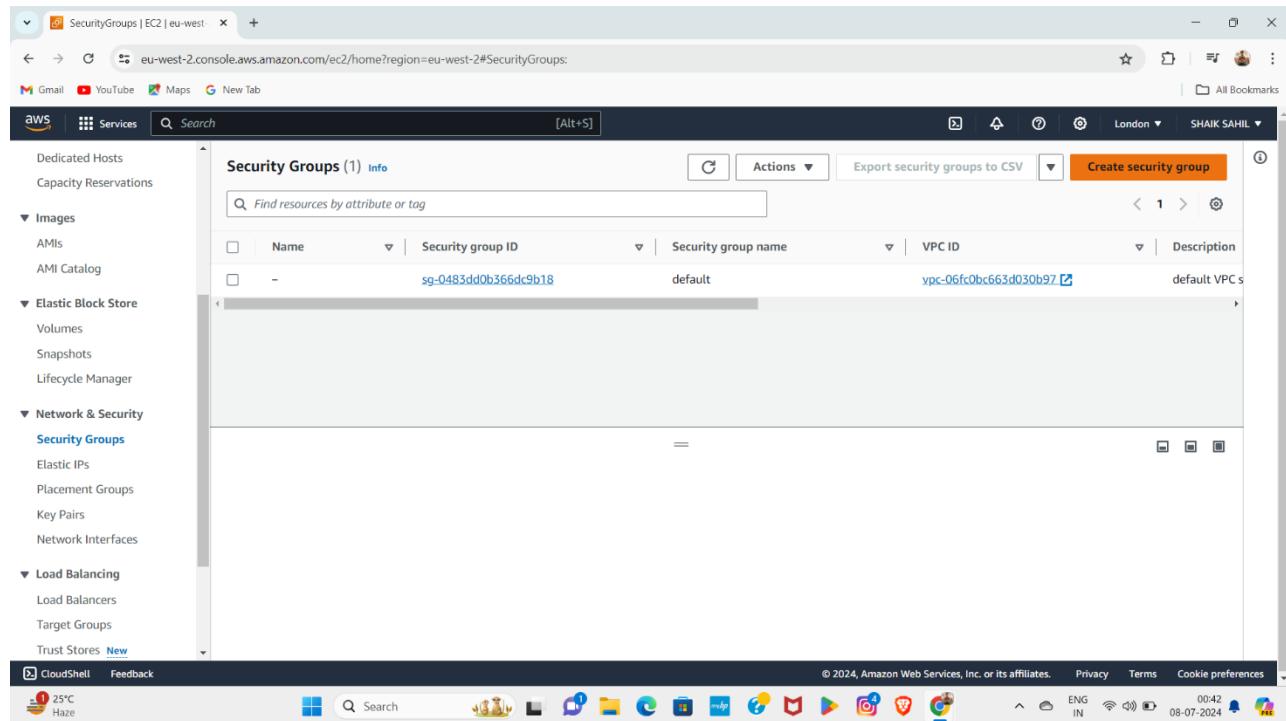
Trainer : Mr . MADHUKAR REDDY

NAME : SHAIK SAHIL

Mail id : sksahil012002@gmail.com

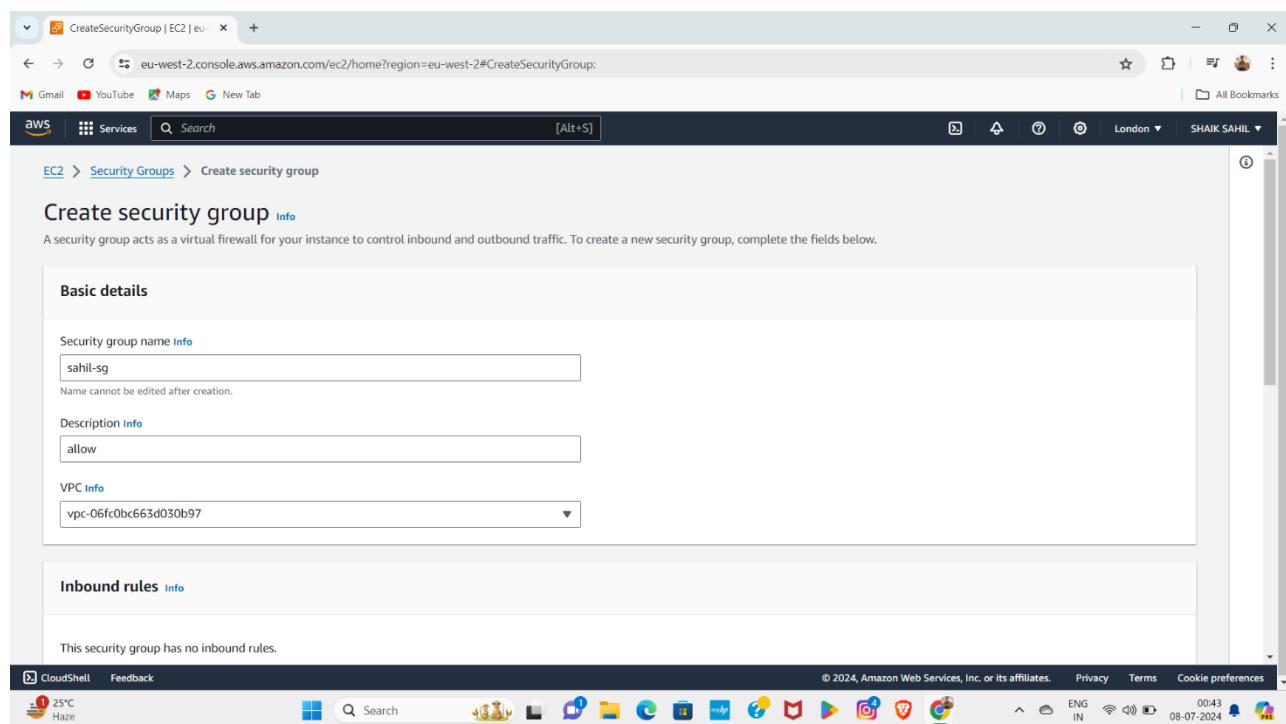
CREATE THREE INSTANCES, INSTALL NGINX AND APPLY APPLICATION LOAD BALANCER (ALB):

Create a security group.



The screenshot shows the AWS EC2 console under the 'Security Groups' section. On the left, a sidebar lists various AWS services like Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security (with 'Security Groups' selected), Load Balancing, and Trust Stores. The main pane displays a table titled 'Security Groups (1) Info' with one row. The row contains the following columns: Name (sg-0483dd0b366dc9b18), Security group ID (sg-0483dd0b366dc9b18), Security group name (default), VPC ID (vpc-06fc0bc663d030b97), and Description (default VPC security group). A search bar at the top of the table allows filtering by attribute or tag. Buttons for 'Actions' and 'Create security group' are also present.

Name a security group.



The screenshot shows the 'Create security group' wizard. The top navigation bar includes links for CloudShell, Feedback, and various system icons. The main content area has a header 'Create security group' with a 'Info' link. Below it, a sub-header 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 'Basic details' section contains three fields: 'Security group name' (set to 'sahil-sg'), 'Description' (set to 'allow'), and 'VPC' (set to 'vpc-06fc0bc663d030b97'). The 'Inbound rules' section is currently empty, with a note: 'This security group has no inbound rules.' The bottom of the screen features a standard Windows taskbar with icons for File Explorer, Task View, Start, and other system utilities.

Add inbound rules.

The screenshot shows the 'CreateSecurityGroup' page in the AWS EC2 console. The 'Inbound rules' section is active, displaying two rules:

Type	Protocol	Port range	Source	Description - optional
SSH	TCP	22	Anywhere (0.0.0.0/0)	
Custom TCP	TCP	80	Anywhere (0.0.0.0/0)	

A yellow warning message at the bottom states: "⚠ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only." An 'Add rule' button is located below the table.

The 'Outbound rules' section is also visible, showing a single rule for 'All traffic' to 'Custom' destination (0.0.0.0/0).

The screenshot shows the 'SecurityGroup | EC2 | eu-west-2' page in the AWS EC2 console. A green success message at the top states: "Security group (sg-0954a163bf9a1cdb1 | sahil-sg) was created successfully".

The main area displays the security group details for 'sg-0954a163bf9a1cdb1 - sahil-sg'. The 'Details' section includes:

Security group name	Security group ID	Description	VPC ID
sahil-sg	sg-0954a163bf9a1cdb1	allow	vpc-06fc0bc665d030b97
Owner	767397761237	Inbound rules count	Outbound rules count
		2 Permission entries	1 Permission entry

The 'Inbound rules' tab is selected, showing two entries:

Type	Protocol	Port range	Source	Description - optional
SSH	TCP	22	Anywhere (0.0.0.0/0)	
Custom TCP	TCP	80	Anywhere (0.0.0.0/0)	

Buttons for 'Actions', 'Manage tags', and 'Edit inbound rules' are available at the bottom of the rules table.

Create a instance.

The screenshot shows the AWS EC2 Instances page. The left sidebar is collapsed. The main content area displays a table with columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IP. A message at the top right says "No instances" and "You do not have any instances in this region". Below the table is a "Launch instances" button. A modal window titled "Select an instance" is open in the foreground.

The screenshot shows the "Launch an instance" wizard. The left sidebar shows the navigation path: EC2 > Instances > Launch an instance. The main content area has several sections: "Name and tags" (with "server1" entered), "Application and OS Images (Amazon Machine Image)" (with a search bar and a note about AMIs), and "Summary" (which includes fields for "Number of instances" set to 1, "Software Image (AMI)" set to "Amazon Linux 2023 AMI 2023.5.2...read more", "Virtual server type (instance type)" set to "t2.micro", "Firewall (security group)" set to "New security group", and "Storage (volumes)" showing "1 volume(s) - 8 GiB"). A "Free tier" callout is visible. At the bottom are "Cancel", "Launch instance" (highlighted in orange), and "Review commands" buttons.

Add subnet and existing security group name.

The screenshot shows the 'Network settings' step of the EC2 instance launch wizard. On the left, under 'VPC - required', a VPC is selected (vpc-06fc0bc663d030b97). A subnet (subnet-0e67d945cf011d8af) is also selected. Under 'Auto-assign public IP', 'Enable' is chosen. In the 'Firewall (security groups)' section, 'Select existing security group' is selected, and 'sahil-sg' is chosen from the dropdown. A note states: '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.' Below this, a note says: 'Additional charges apply when outside of free tier allowance'. A 'Common security groups' section shows 'sahil-sg sg-0954a163bf9a1cdb1' selected. A note at the bottom says: 'Security groups that you add or remove here will be added to or removed from all your network interfaces.'

Summary

Number of instances: 1

Software Image (AMI): Amazon Linux 2023 AMI 2023.5.2... [read more](#)

Virtual server type (instance type): t2.micro

Firewall (security group): sahil-sg

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

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro) in the Regions in which you launch instances.

Launch instance

The screenshot shows the 'Success' step of the EC2 instance launch wizard. It displays a green banner stating: 'Successfully initiated launch of instance (i-0a83a37c35c152408)'. Below this, there is a 'Launch log' link. The 'Next Steps' section contains several cards:

- Create billing and free tier usage alerts**: To manage costs and avoid surprise bills, set up email notifications for billing and free tier usage thresholds. Includes a 'Create billing alerts' button.
- Connect to your instance**: Once your instance is running, log in from your local computer. Includes a 'Connect to instance' button and a 'Learn more' link.
- Connect an RDS database**: Configure the connection between an EC2 instance and a database to allow traffic flow between them. Includes a 'Connect an RDS database' button and a 'Create a new RDS database' link.
- Create EBS snapshot policy**: Create a policy that automates the creation, retention, and deletion of EBS snapshots. Includes a 'Create EBS snapshot policy' button.

Next Steps

What would you like to do next with this instance, for example "create alarm" or "create backup"

The screenshot shows the AWS EC2 Instances page. The left sidebar is collapsed. The main area displays a table titled 'Instances (1) Info' with one row. The row contains the following columns: Name (server1), Instance ID (i-0a83a37c35c152408), Instance state (Running), Status check (Initializing), Alarm status (View alarms +), Availability Zone (eu-west-2a), and Public IP (ec2-18-). Below the table, a modal window titled 'Select an instance' is open, showing the same single instance entry.

Reapeat the same process two times to have three instances.

The screenshot shows the AWS EC2 Instances page after three instances have been created. The left sidebar is collapsed. The main area displays a table titled 'Instances (3) Info' with three rows. The rows contain the following information:

Name	Instance ID	Instance state	Status check	Alarm status	Availability Zone	Public IP
server2	i-0b20681c0571b56e	Running	Initializing	View alarms +	eu-west-2b	ec2-18-
server1	i-0a83a37c35c152408	Running	Initializing	View alarms +	eu-west-2a	ec2-18-
server3	i-0892e898a4d023eb9	Running	-	View alarms +	eu-west-2c	ec2-18-

Below the table, a modal window titled 'Select an instance' is open, showing the three instances listed.

Connect to instance and copy the command from SSH client to check the instance if it is working or not on OS.

```

root@ip-172-31-18-9:~/share/nginx/html
$ ssh -i "lb-key.pem" ec2-user@ec2-18-135-100-200.eu-west-2.compute.amazonaws.com
The authenticity of host 'ec2-18-135-100-200.eu-west-2.compute.amazonaws.com (18.135.100.200)' can't be established.
ED25519 key fingerprint is SHA256:zH23KjM9669ywo7YHnbwSPMIVkqH9ASVkjHBR+1U.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
warning: Permanently added 'ec2-18-135-100-200.eu-west-2.compute.amazonaws.com' (ED25519) to the list of known hosts.

Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

[ec2-user@ip-172-31-18-9 ~]$ sudo -i
[ec2-user@ip-172-31-18-9 ~]$ yum update -y && yum install nginx -y && cd /usr/share/nginx/html
Last metadata expiration check: 0:12:16 ago on Sun Jul 7 19:19:41 2024.
Dependencies resolved.
Nothing to do.
Complete!
Last metadata expiration check: 0:12:16 ago on Sun Jul 7 19:19:41 2024.
Dependencies resolved.

Transaction Summary
Install 7 Packages

Total download size: 1.0 M
Installed size: 1.4 M
Downloaded packages:
(1/7): generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch.rpm 323 kB/s | 19 kB 00:00
(2/7): gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64.rpm 2.0 MB/s | 308 kB 00:00
(3/7): nginx-1.24.0-1.amzn2023.0.2.x86_64.rpm 25 MB/s | 32 kB 00:00
(4/7): nginx-core-1.24.0-1.amzn2023.0.2.x86_64.rpm 78 kB/s | 586 kB 00:00
(5/7): nginx-filesystem-1.24.0-1.amzn2023.0.2.noarch.rpm 473 kB/s | 9.1 kB 00:00
(6/7): nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch.rpm 1.1 MB/s | 21 kB 00:00
(7/7): nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch.rpm 6.3 MB/s | 1.0 MB 00:00

total
Running transaction check
Transaction check succeeded.
Running transaction test

```

```
root@ip-172-31-18-9:/usr/share/nginx/html
nginx
Installing dependencies:
  generic-logos-httdp
  gperftools-libs
  libunwind
  nginx-core
  nginx-fsletem
  nginx-mimetypes
x86_64                               1:1.24.0-1.amzn2023.0.2
noarch                                18.0.0-12.amzn2023.0.3
                                         2.9.1-1.amzn2023.0.3
                                         1.4.0-5.amzn2023.0.2
                                         1:1.24.0-1.amzn2023.0.2
                                         1:1.24.0-1.amzn2023.0.2
                                         2.1.49-3.amzn2023.0.3

Transaction Summary
Install 7 Packages

Total download size: 1.0 M
Installed size: 3.4 M
Downloading Packages:
(1/7): generic-logos-httdp-18.0.0-12.amzn2023.0.3.noarch.rpm 323 kB/s | 19 kB 00:00
(2/7): gperftools-lbs-1.4.0-5.amzn2023.0.2.x86_64.rpm        47 kB/s | 30 kB 00:00
(3/7): libunwind-1.4.0-5.amzn2023.0.2.x86_64.rpm           2.0 MB/s | 32 kB 00:00
(4/7): nginx-core-1.24.0-1.amzn2023.0.2.x86_64.rpm         25 MB/s | 586 kB 00:00
(5/7): libunwind-1.4.0-5.amzn2023.0.2.x86_64.rpm           713 kB/s | 66 kB 00:00
(6/7): nginx-fsletem-1.24.0-1.amzn2023.0.2.noarch.rpm      473 kB/s | 9.1 kB 00:00
(7/7): nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch.rpm    1.1 MB/s | 21 kB 00:00
Total                                         6.3 MB/s | 1.0 MB 00:00

Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing                                           1/1
    Running scriptlet: nginx-fsletem-1:1.24.0-1.amzn2023.0.2.noarch
  Installing : nginx-fsletem-1:1.24.0-1.amzn2023.0.2.noarch
  Installing : libunwind-1.4.0-5.amzn2023.0.2.x86_64
  Installing : gperftools-lbs-1.4.0-5.amzn2023.0.2.x86_64
  Installing : nginx-core-1:1.24.0-1.amzn2023.0.2.x86_64
  Installing : generic-logos-httdp-18.0.0-12.amzn2023.0.3.noarch
  Installing : nginx-11.24.0-1.amzn2023.0.2.x86_64
  Running scriptlet: nginx-fsletem-1:1.24.0-1.amzn2023.0.2.noarch
  Verifying   : generic-logos-httdp-18.0.0-12.amzn2023.0.3.noarch
  Verifying   : libunwind-1.4.0-5.amzn2023.0.2.x86_64
  Verifying   : nginx-11.24.0-1.amzn2023.0.2.x86_64
  Verifying   : nginx-core-1:1.24.0-1.amzn2023.0.2.x86_64
  Verifying   : nginx-fsletem-1:1.24.0-1.amzn2023.0.2.noarch
  Verifying   : nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch

Installed:
  generic-logos-httdp-18.0.0-12.amzn2023.0.3.noarch
  gperftools-lbs-1.4.0-5.amzn2023.0.2.x86_64
  libunwind-1.4.0-5.amzn2023.0.2.x86_64
  nginx-core-1:1.24.0-1.amzn2023.0.2.x86_64
  nginx-fsletem-1:1.24.0-1.amzn2023.0.2.noarch
  nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch

Completed!
[root@ip-172-31-18-9 html]# rm index.html
rm: remove regular file 'index.html'? yes
[root@ip-172-31-18-9 html]# vi index.html
[root@ip-172-31-18-9 html]# systemctl restart nginx
[root@ip-172-31-18-9 html]#
```

Copy IP address form EC2 instance.

The screenshot shows the AWS EC2 'Connect to instance' interface. At the top, the URL is eu-west-2.console.aws.amazon.com/ec2/home?region=eu-west-2#ConnectToInstance:instanceId=i-0a83a37c35c152408. The AWS logo is in the top-left corner, followed by 'Services'. A search bar contains '[Alt+S]'. On the right, there are icons for star, download, and more, along with 'London' and 'SHAIK SAHIL'.

The main navigation path is EC2 > Instances > i-0a83a37c35c152408 > Connect to instance.

Connect to instance Info

Connect to your instance i-0a83a37c35c152408 (server1) using any of these options

EC2 Instance Connect Session Manager | SSH client | EC2 serial console

Instance ID: i-0a83a37c35c152408 (server1)

Connection Type:

- Connect using EC2 Instance Connect**
Connect using the EC2 Instance Connect browser-based client with a public IP address.
- Connect using EC2 Instance Connect Endpoint**
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

Public IP address copied (with a green checkmark icon)

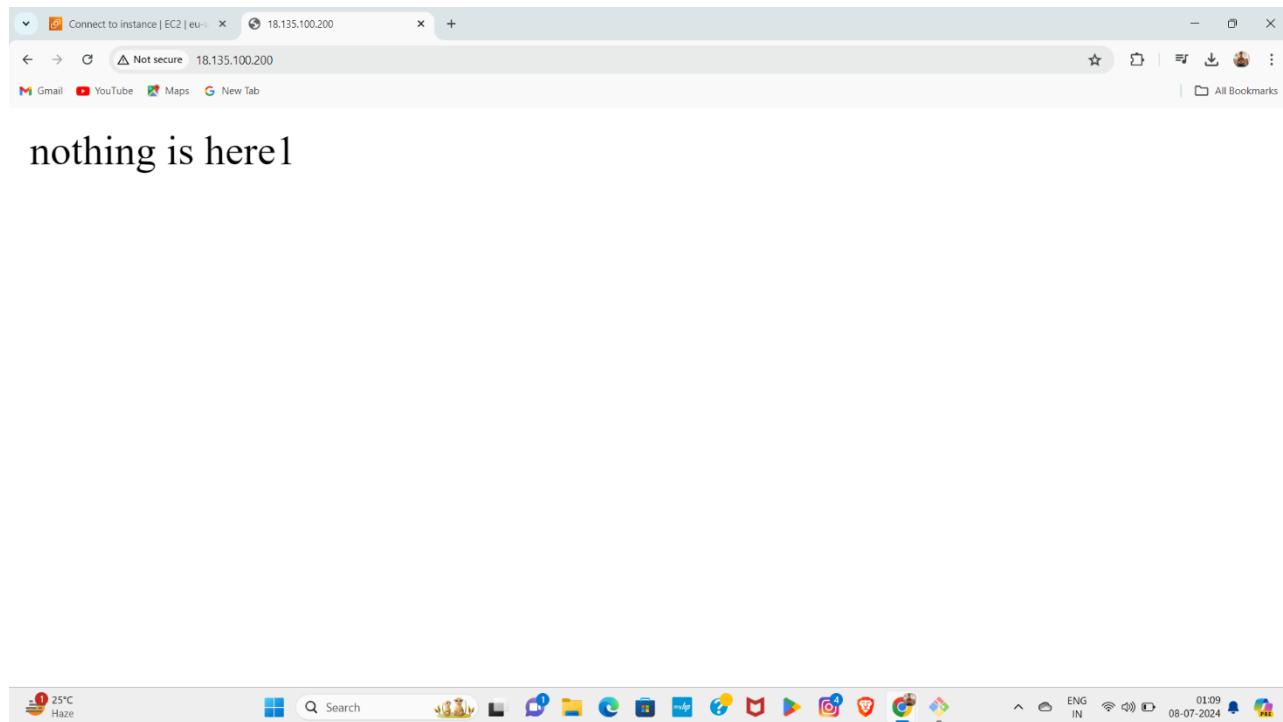
IP Address: 18.135.100.200

Username: ec2-user

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.

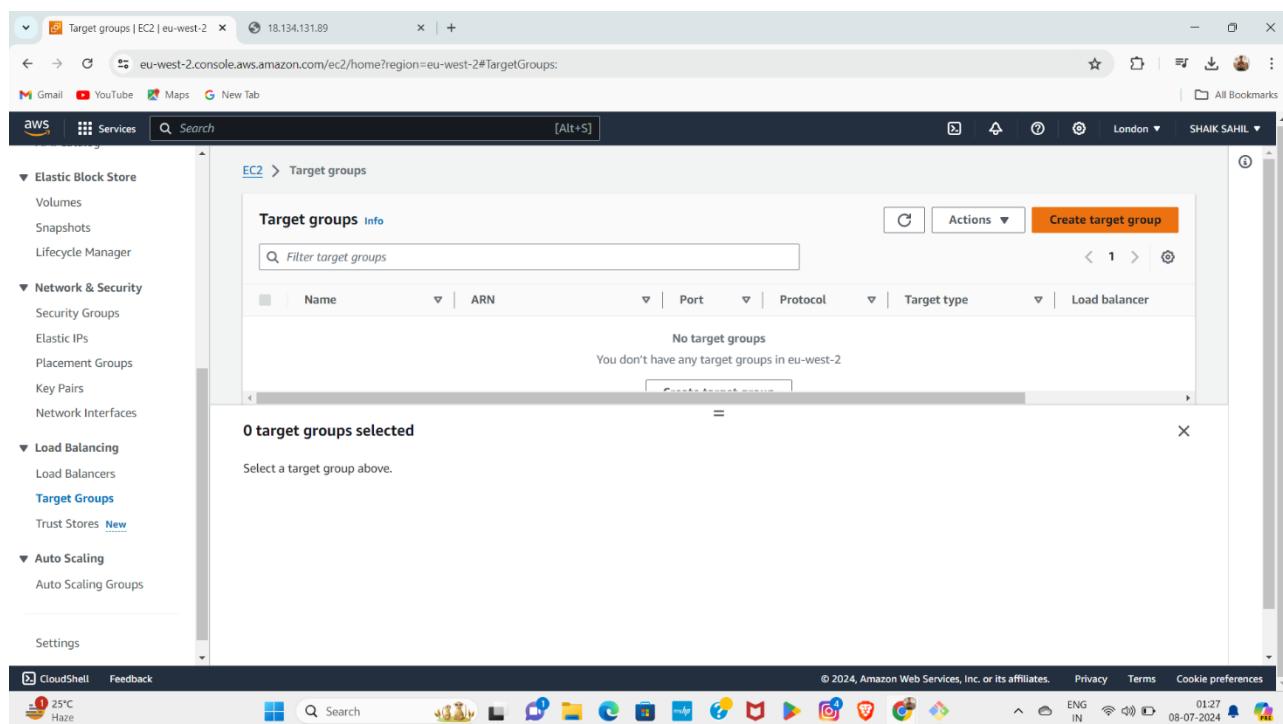
At the bottom, there are links for CloudShell, Feedback, and various AWS services like Lambda, S3, CloudWatch, and others. The footer includes copyright information for 2024, privacy terms, and cookie preferences, along with system status icons for battery, signal, and network.

Paste it on web browser to check the data in instance.



Repeat the same process with other two instances.

Create target group.



Select instances target type.

The screenshot shows the 'Specify group details' step of the 'Create target group' wizard. Under 'Basic configuration', the 'VPC' section is expanded, showing 'vpc-12345678'. Below it, 'Security groups' lists 'sg-12345678'. The 'Choose a target type' section has three options: 'Instances' (selected), 'IP addresses', and 'Lambda function'. The 'Instances' section includes a note: 'Your load balancer routes requests to the targets in a target group and performs health checks on the targets.' and a bulleted list: 'Supports load balancing to instances within a specific VPC.', 'Facilitates the use of [Amazon EC2 Auto Scaling](#) to manage and scale your EC2 capacity.' The 'IP addresses' section includes a bulleted list: 'Supports load balancing to VPC and on-premises resources.', 'Facilitates routing to multiple IP addresses and network interfaces on the same instance.', 'Offers flexibility with microservice based architectures, simplifying inter-application communication.', 'Supports IPv6 targets, enabling end-to-end IPv6 communication, and IPv4-to-IPv6 NAT.' The 'Lambda function' section includes a bulleted list: 'Facilitates routing to a single Lambda function.', 'Accessible to Application Load Balancers only.'

Name a target group.

The screenshot shows the 'Specify group details' step of the 'Create target group' wizard. Under 'Basic configuration', the 'VPC' section is expanded, showing 'vpc-12345678'. Below it, 'Security groups' lists 'sg-12345678'. The 'Choose a target type' section has three options: 'Instances' (selected), 'IP addresses', and 'Lambda function'. The 'Lambda function' section includes a bulleted list: 'Facilitates routing to a single Lambda function.', 'Accessible to Application Load Balancers only.' The 'Target group name' field contains 'sahil-tg'. A note below says: 'A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.' The 'Protocol : Port' section shows 'HTTP' selected with port 80. A note below says: 'Choose a protocol for your target group that corresponds to the Load Balancer type that will route traffic to it. Some protocols now include anomaly detection for the targets and you can set mitigation options once your target group is created. This choice cannot be changed after creation.' The 'IP address type' section shows 'IPv4' selected. A note below says: 'Each instance has a default network interface (eth0) that is assigned the primary private IPv4 address. The instance's primary private IPv4 address is the one that will be applied to the target.' The 'IPv6' option is also present with a note: 'Each instance you register must have an assigned primary IPv6 address. This is configured on the instance's default network interface (eth0). Learn more [\[link\]](#)'.

Regeister three instances in targets.

The screenshot shows the 'Register targets' step of the 'Create target group' wizard. On the left, there are two tabs: 'Specify group details' (selected) and 'Register targets'. The main area is titled 'Available instances (3/3)' and lists three instances: server3, server2, and server1, all in the 'Running' state and associated with the security group 'sahil-sg'. Below the instance list, a section titled 'Ports for the selected instances' shows the port number '80' entered. A button 'Include as pending below' is visible. At the bottom, a note says '3 selected'.

Click on create target group.

The screenshot shows the 'Review targets' step of the 'Create target group' wizard. It displays the 'Targets (3)' table with the same three instances as before. The table includes columns for Instance ID, Name, Port, State, Security groups, Zone, Private IPv4 address, and Subnet ID. A note at the top right says '3 selections are now pending below. Include more or register targets when ready.' At the bottom, there are buttons for 'Cancel', 'Previous', and a prominent orange 'Create target group' button.

The screenshot shows the AWS EC2 Target group details page. A success message at the top states: "Successfully created the target group: sahil-tg. Anomaly detection is automatically applied to all registered targets. Results can be viewed in the Targets tab." The navigation path is EC2 > Target groups > sahil-tg. The target group name is "sahil-tg". The "Details" section shows the following configuration:

Target type	Protocol : Port	Protocol version
Instance	HTTP: 80	HTTP1
IP address type	Load balancer	VPC
IPv4	None associated	vpc-06fc0bc663d030b97

Statistics for targets:

Total targets	3	Healthy	0	Unhealthy	0
Unused	3	Initial	0	Draining	0
0 Anomalous					

A section titled "Distribution of targets by Availability Zone (AZ)" is present, with a note: "Select values in this table to see corresponding filters applied to the Registered targets table below."

Bottom navigation bar: CloudShell, Feedback, © 2024, Amazon Web Services, Inc. or its affiliates., Privacy, Terms, Cookie preferences.

Create a load balancer.

The screenshot shows the AWS EC2 Load balancers page. The navigation path is EC2 > Load balancers. A message at the top states: "Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic." Below is a table to manage load balancers:

Name	DNS name	State	VPC ID	Availability Zones	Type
No load balancers You don't have any load balancers in eu-west-2					

Below the table, it says "0 load balancers selected" and "Select a load balancer above." The bottom navigation bar includes CloudShell, Feedback, © 2024, Amazon Web Services, Inc. or its affiliates., Privacy, Terms, Cookie preferences.

Create a application load balancer.

The screenshot shows the 'Create Application Load Balancer' wizard on the AWS EC2 console. The 'Basic configuration' step is selected. The 'Load balancer name' field contains 'sahil-lb'. The 'Scheme' section has 'Internet-facing' selected, which routes requests from the internet to targets. The 'Mappings' section lists three subnets: 'eu-west-2a (euw2-az2)', 'eu-west-2b (euw2-az3)', and 'eu-west-2c (euw2-az1)'. Each subnet is assigned by AWS. The browser status bar at the bottom indicates the URL is 18.134.131.89 and the page is from eu-west-2.console.aws.amazon.com.

Add subnets in mapping.

The screenshot shows the 'Network mapping' configuration screen. It lists three subnets under 'Mappings': 'eu-west-2a (euw2-az2)', 'eu-west-2b (euw2-az3)', and 'eu-west-2c (euw2-az1)'. Each subnet is assigned by AWS. The browser status bar at the bottom indicates the URL is 18.134.131.89 and the page is from eu-west-2.console.aws.amazon.com.

Add security group that created before.

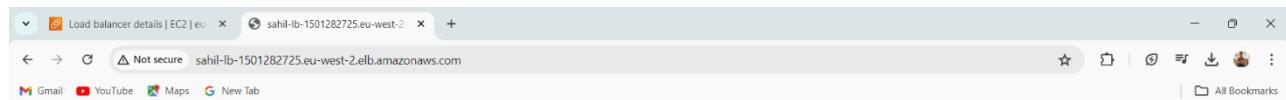
The screenshot shows the AWS Lambda console with the title "Create application load balance". In the "Security groups" section, a security group named "sahil-sg" is selected. Below it, the "Listeners and routing" section shows a listener for port 80 forwarding traffic to a target group named "sahil-tg".

Copy the DNS name from load balancer.

The screenshot shows the AWS EC2 Load Balancers console with the title "Load balancer details | EC2 | eu". A success message states "Successfully created load balancer: sahil-lb". The "Details" section shows the load balancer ARN: arn:aws:elasticloadbalancing:eu-west-2:767397761237:loadbalancer/app/sahil-lb/6b936c6961b0abb4. A tooltip indicates the DNS name has been copied: "sahil-lb-1501282725.eu-west-2.elb.amazonaws.com (A Record)".

Check the copied link in web browser and do refresh and it shows the result be like-

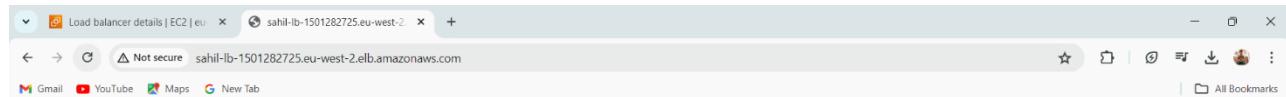
First refresh.



nothing here2



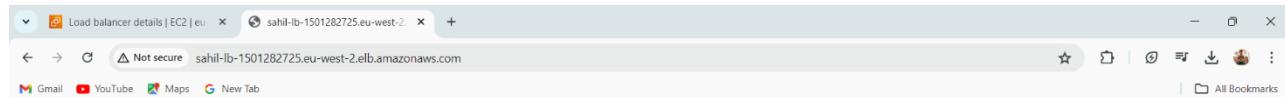
Second refresh.



nothing here3



Third refresh.



nothing is here1

