



## ASSIGNMENT – 02

COURSE : AWS DEVOPS

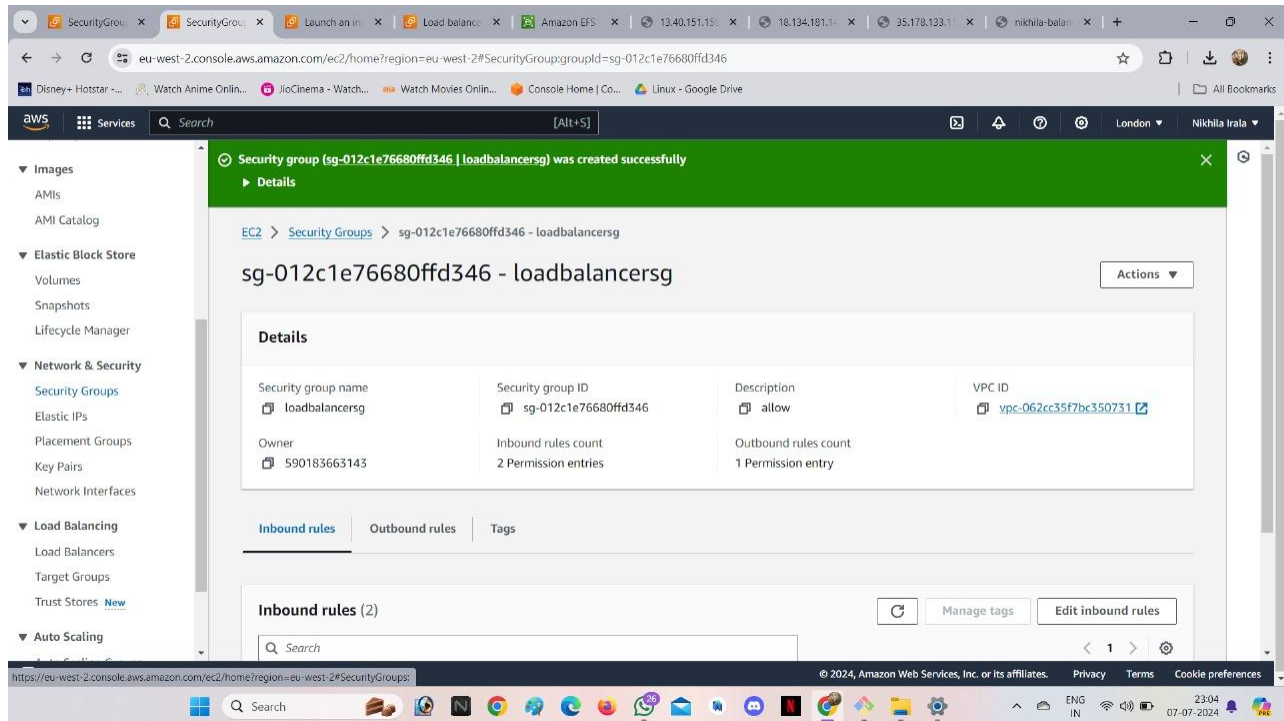
Trainer : Mr . MADHUKAR REDDY

NAME : NIKHILA

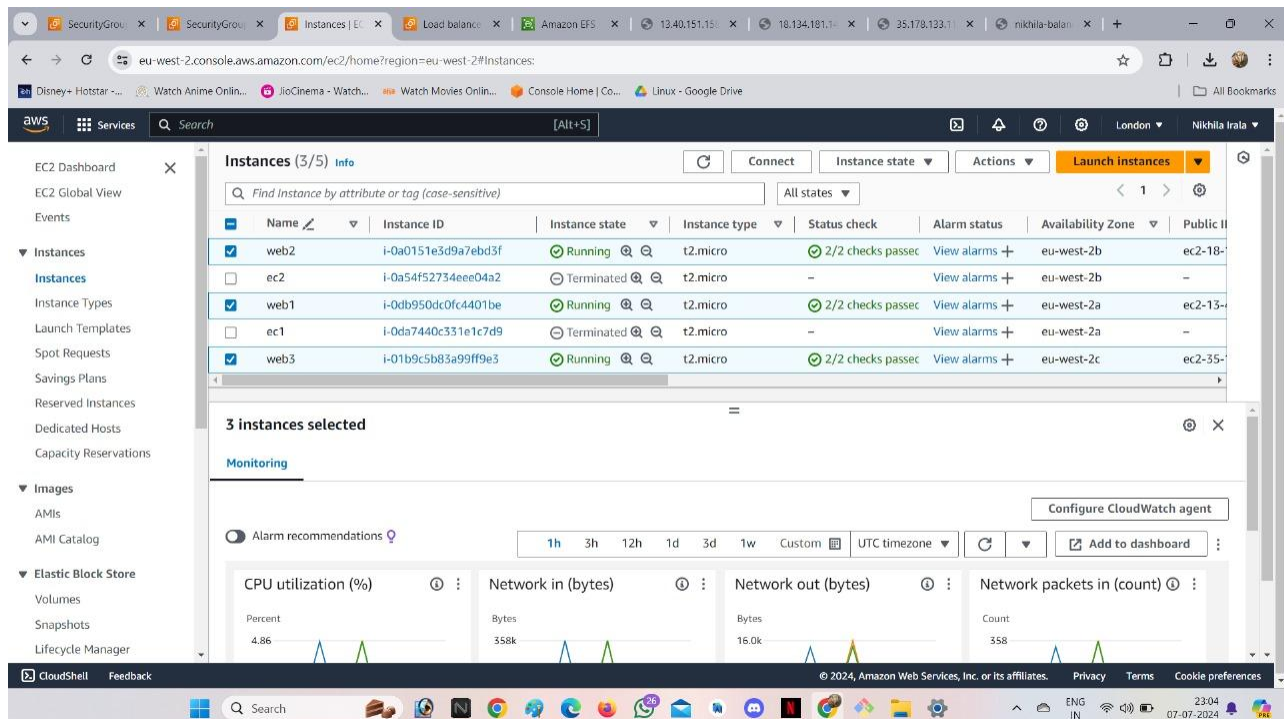
Mail id : [Nikhilairala@gmail.com](mailto:Nikhilairala@gmail.com)

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

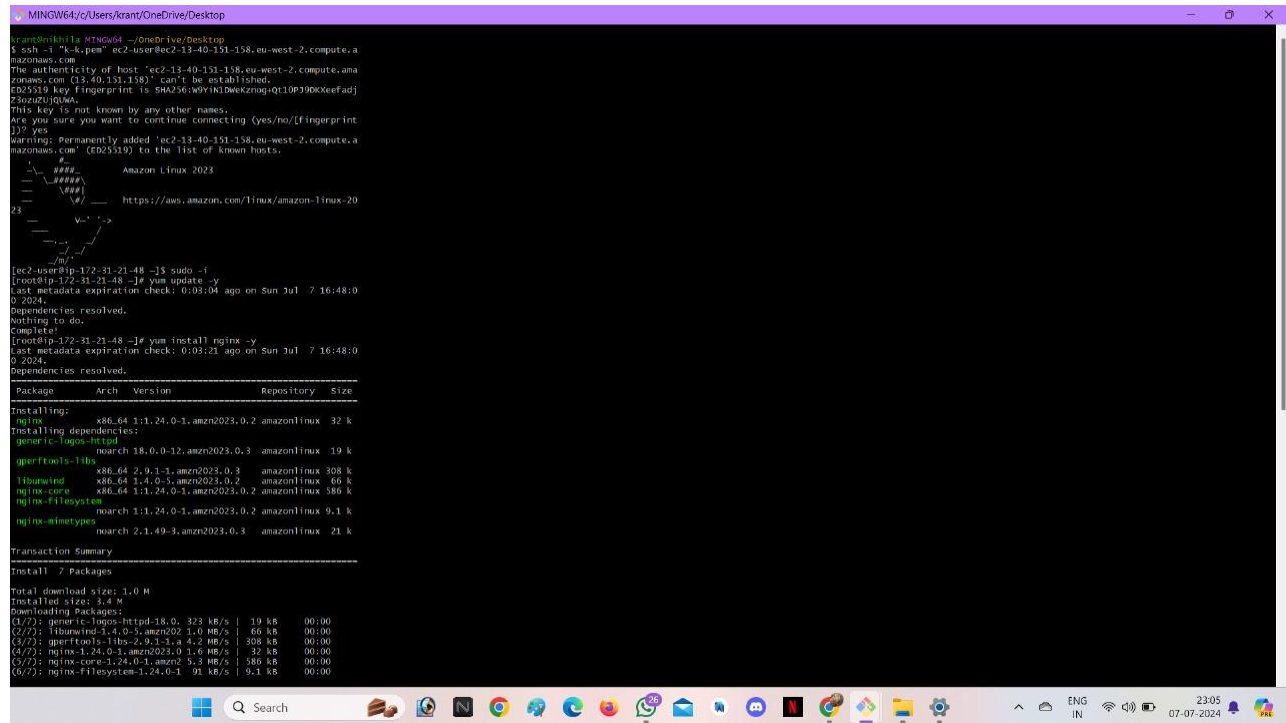
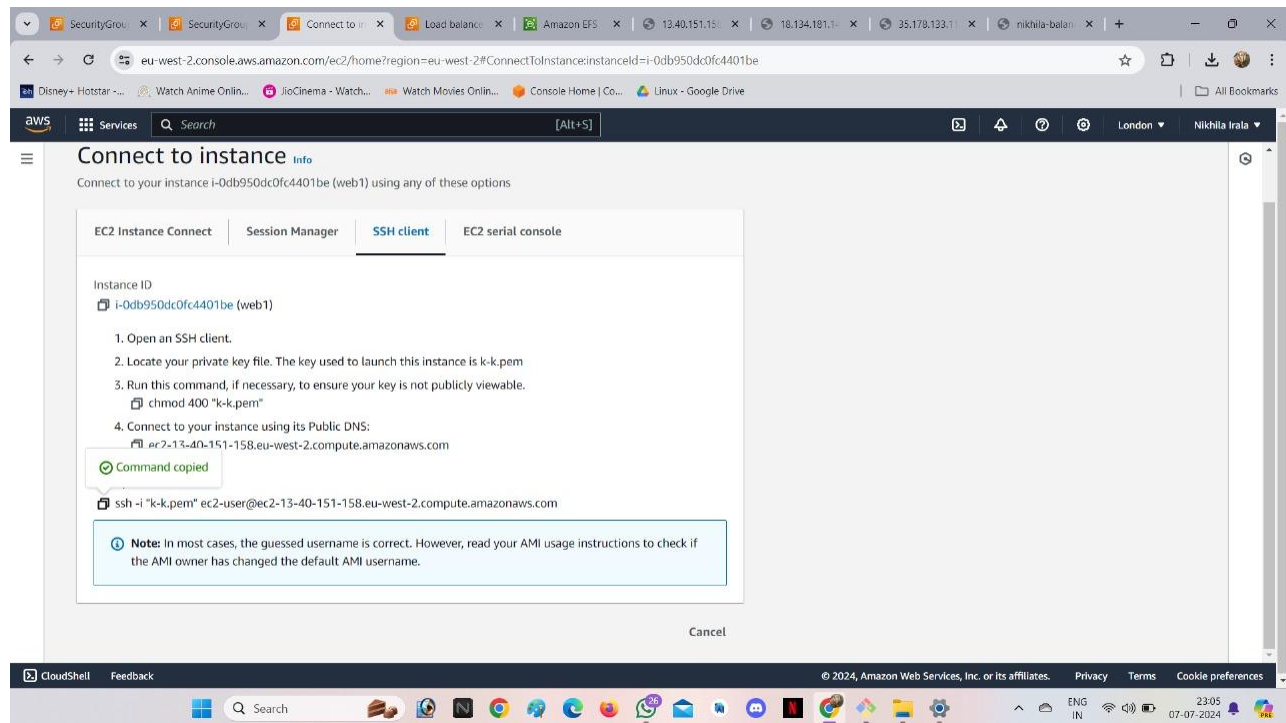
Create a security group.



Create three instances.



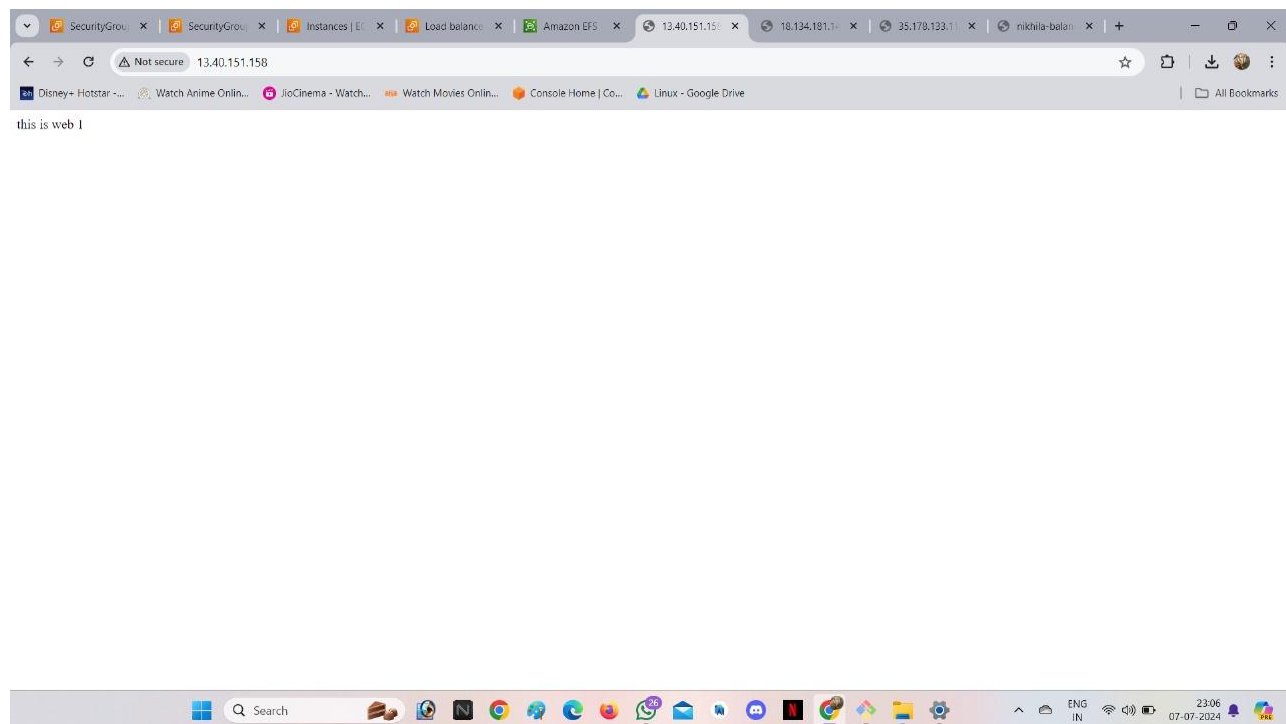
Connect to instance and copy the command from SSH client to check the instance if it is working or not on OS.



```
MINGW64/c:/Users/krantOneDrive/Desktop
nginx-core-1.24.0-1.amzn2023.0.2.amazonlinux x86_64
nginx-filesystem
noarch 1:1.24.0-1.amzn2023.0.2.amazonlinux 9.1 k
nginx-minetypes
noarch 2:1.49-3.amzn2023.0.3.amazonlinux 21 k

Transaction Summary
Install 7 Packages
Total download size: 3.0 MB
Installed size: 3.4 MB
Downloading Packages:
(1/7): generic-logos-httpd-18.0.323.kb/s | 19 kB | 00:00
(2/7): libbunwind-1.4.0-5.amzn2023.0.2.x86_64 66 kB | 00:00
(3/7): gperftools-libs-2.9.1-3.a 4.2 MB/s | 308 kB | 00:00
(4/7): nginx-1.24.0-1.amzn2023.0.2.x86_64 3.6 MB/s | 32 kB | 00:00
(5/7): nginx-core-1.24.0-1.amzn2023.0.2.x86_64 5.1 MB/s | 369 kB | 00:00
(6/7): nginx-filesystem-1.24.0-1.amzn2023.0.2.x86_64 9.1 kB | 00:00
(7/7): nginx-minetypes-2.1.49-3.amzn2023.0.3.noarch 21 kB | 00:00
Total 4.3 MB/s | 3.0 MB | 00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing : 1/7
Running scriptlet: nginx-filesystem-1:1.24.0-1.amzn2023.0.2.x86_64 1/7
Installing : nginx-filesystem-1:1.24.0-1.amzn2023.0.2.x86_64 1/7
Installing : nginx-minetypes-2:1.49-3.amzn2023.0.3.noarch 2/7
Installing : libbunwind-1.4.0-5.amzn2023.0.2.x86_64 3/7
Installing : gperftools-libs-2.9.1-3.a 4/7
Installing : nginx-core-1:1.24.0-1.amzn2023.0.2.x86_64 5/7
Installing : generic-logos-httpd-18.0.0-12.amzn2023.0.2.x86_64 6/7
Installing : nginx-1:1.24.0-1.amzn2023.0.2.x86_64 7/7
Running scriptlet: nginx-1:1.24.0-1.amzn2023.0.2.x86_64 7/7
Verifying : generic-logos-httpd-18.0.0-12.amzn2023.0.2.x86_64 1/7
Verifying : gperftools-libs-2.9.1-3.a 2/7
Verifying : libbunwind-1.4.0-5.amzn2023.0.2.x86_64 3/7
Verifying : nginx-1:1.24.0-1.amzn2023.0.2.x86_64 4/7
Verifying : nginx-core-1:1.24.0-1.amzn2023.0.2.x86_64 5/7
Verifying : nginx-filesystem-1:1.24.0-1.amzn2023.0.2.x86_64 6/7
Verifying : nginx-minetypes-2:1.49-3.amzn2023.0.3.noarch 7/7
Installed:
generic-logos-httpd-18.0.0-12.amzn2023.0.2.x86_64
gperftools-libs-2.9.1-3.a
libbunwind-1.4.0-5.amzn2023.0.2.x86_64
nginx-1:1.24.0-1.amzn2023.0.2.x86_64
nginx-core-1:1.24.0-1.amzn2023.0.2.x86_64
nginx-filesystem-1:1.24.0-1.amzn2023.0.2.x86_64
nginx-minetypes-2:1.49-3.amzn2023.0.3.noarch
complete!
[root@ip-172-31-21-48 ~]# cd /usr/share/nginx/html
[root@ip-172-31-21-48 html]# rm index.html
rm: remove regular file 'index.html'? yes
[root@ip-172-31-21-48 html]# vi index.html
[root@ip-172-31-21-48 html]# systemctl restart nginx
[root@ip-172-31-21-48 html]# client_loop: send disconnect: Connection reset by peer
krant@nikhila: MINGW64 ~/OneDrive/Desktop
$
```

Copy IP address from EC2 instance and check in web browser.



## Create a target group.

The screenshot shows the AWS Management Console for the eu-west-2 region. The 'Target groups' page is active, displaying a table with one target group: 'nikhila-target'. The details modal for 'nikhila-target' is open, showing the following information:

Target type	Protocol : Port	Protocol version	VPC
Instance	HTTP: 80	HTTP1	vpc-062cc35f7bc350731

## Create a load balancer.

The screenshot shows the AWS Management Console for the eu-west-2 region. The 'Load balancers' page is active, displaying a table with one load balancer: 'nikhila-balancer'. The details modal for 'nikhila-balancer' is open, showing the following information:

Load balancer type	Status	VPC	Load balancer IP address type
Application	Active	vpc-062cc35f7bc350731	IPv4



Copy DNS name and check it on web browser and Do refresh it.

The screenshot shows the AWS Management Console interface. On the left, there's a navigation menu with categories like 'Elastic Block Store', 'Network & Security', 'Load Balancing', and 'Auto Scaling'. The main content area displays 'Load balancers (1/1)' with a table listing the 'nikhila-balancer'. A modal window titled 'Load balancer: nikhila-balancer' is open, showing details such as the 'Load balancer ARN' and a list of subnets. A green notification bubble indicates 'DNS name copied'.

Name	DNS name	State	VPC ID	Availability Zones	Type
nikhila-balancer	nikhila-balancer-503736414.eu-west-2.elb.amazonaws.com	Active	vpc-062cc35f7bc350731	3 Availability Zones	application

Load balancer: nikhila-balancer

Load balancer ARN  
arn:aws:elasticloadbalancing:eu-west-2:590183663143:loadbalancer/app/nikhila-balancer/27f692c8f0216249

Subnets:  
west-2c (euw2-az1)  
subnet-027f9e55ed61edfd3 eu-west-2b (euw2-az3)  
subnet-07da943a6e0ef948f eu-west-2a (euw2-az2)

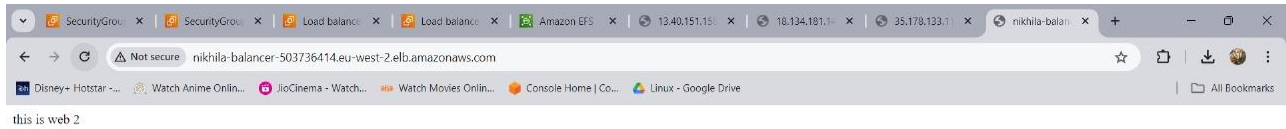
DNS name copied  
nikhila-balancer-503736414.eu-west-2.elb.amazonaws.com (A Record)

First refresh.

The screenshot shows a web browser window with the address bar displaying 'nikhila-balancer-503736414.eu-west-2.elb.amazonaws.com'. The browser status bar indicates 'Not secure'. The page content is blank, and the browser's taskbar at the bottom shows various application icons and the system clock.

this is web 1

Second refresh.



Third refresh.

