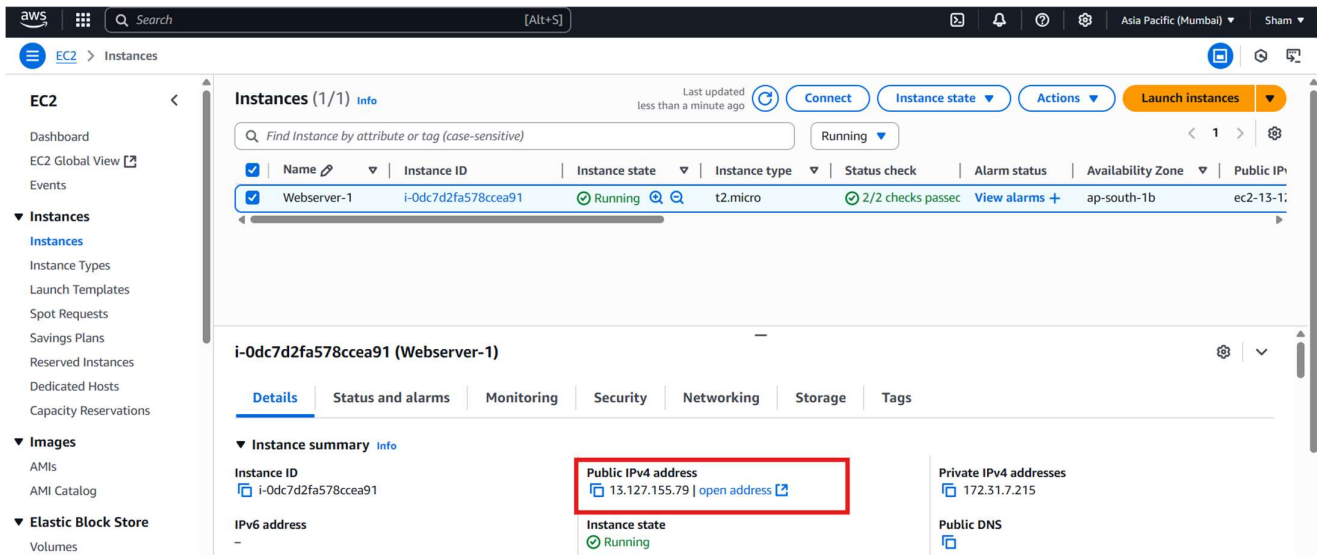
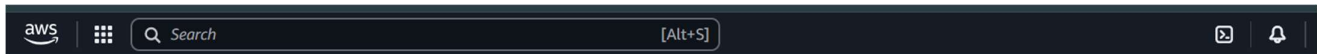


Create Two Instances with website and Access on Server1 with Root Domain and Access on Server 2 With Subdomain

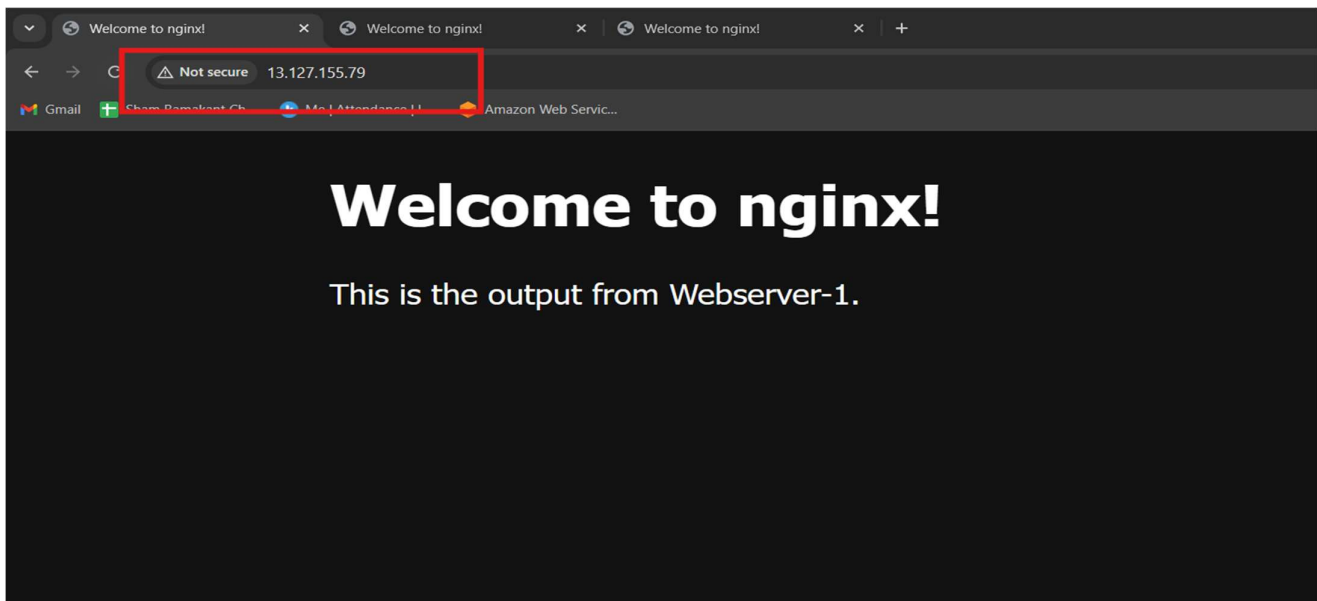
1. Create Server 1 and hosted website.



The screenshot shows the AWS Management Console for EC2 Instances. The instance 'Webserver-1' (ID: i-0dc7d2fa578ccea91) is in a 'Running' state. The public IPv4 address 13.127.155.79 is highlighted with a red box. The instance summary shows the instance is running and has a public IPv4 address.



```
root@ip-172-31-7-215:~# cd /var/www/html/
root@ip-172-31-7-215:/var/www/html# ls
index.nginx-debian.html
root@ip-172-31-7-215:/var/www/html# vi index.nginx-debian.html
root@ip-172-31-7-215:/var/www/html#
```



The screenshot shows a web browser window with the address bar displaying '13.127.155.79'. The page content displays 'Welcome to nginx!' and 'This is the output from Webserver-1.' The browser's address bar and the page content are highlighted with a red box.

2. Create Server 2 and hosted website.

The screenshot shows the AWS Management Console for EC2 instances. The left sidebar contains navigation links for EC2, Images, and Elastic Block Store. The main content area displays a list of instances. Two instances are shown: Webserver-1 (i-0dc7d2fa578ccea91) and Webserver-2 (i-0850a9258a3b71bd8). Webserver-2 is selected, and its details are shown below. The 'Public IPv4 address' is highlighted with a red box and is 13.127.59.162. The 'Private IPv4 addresses' are 172.31.47.223.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
Webserver-1	i-0dc7d2fa578ccea91	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1b	ec2-13-1...
Webserver-2	i-0850a9258a3b71bd8	Running	t2.micro	Initializing	View alarms +	ap-south-1a	ec2-13-1...

i-0850a9258a3b71bd8 (Webserver-2)

Instance summary

Instance ID: i-0850a9258a3b71bd8

Public IPv4 address: 13.127.59.162 | [open address](#)

Private IPv4 addresses: 172.31.47.223

```
root@ip-172-31-47-223:~# vi /var/www/html/index.nginx-debian.html
root@ip-172-31-47-223:~# vi /var/www/html/index.nginx-debian.html
root@ip-172-31-47-223:~# vi /var/www/html/index.nginx-debian.html
root@ip-172-31-47-223:~#
```

The screenshot shows a web browser window with the address bar displaying 'Not secure 13.127.59.162'. The page content displays 'Welcome to nginx!' and 'This is the output from Webserver-2.' The address bar and the message are highlighted with a red box.

3. Mapping and Access on Server1 with Root Damin: globaltest24.online

MANAGE DNS RECORDS

For globaltest24.online

ALL RECORDS | A | AAAA | MX | CNAME | NS | TXT | SRV | SOA

ADD A RECORD

DOMAIN/SUB-DOMAIN	IN	A	IPv4 ADDRESS*
TTL (Time to Live) Choose TTL			

These changes will take 4-6 hours to come into effect.

+ Add Another Record

SAVE RECORD(S)

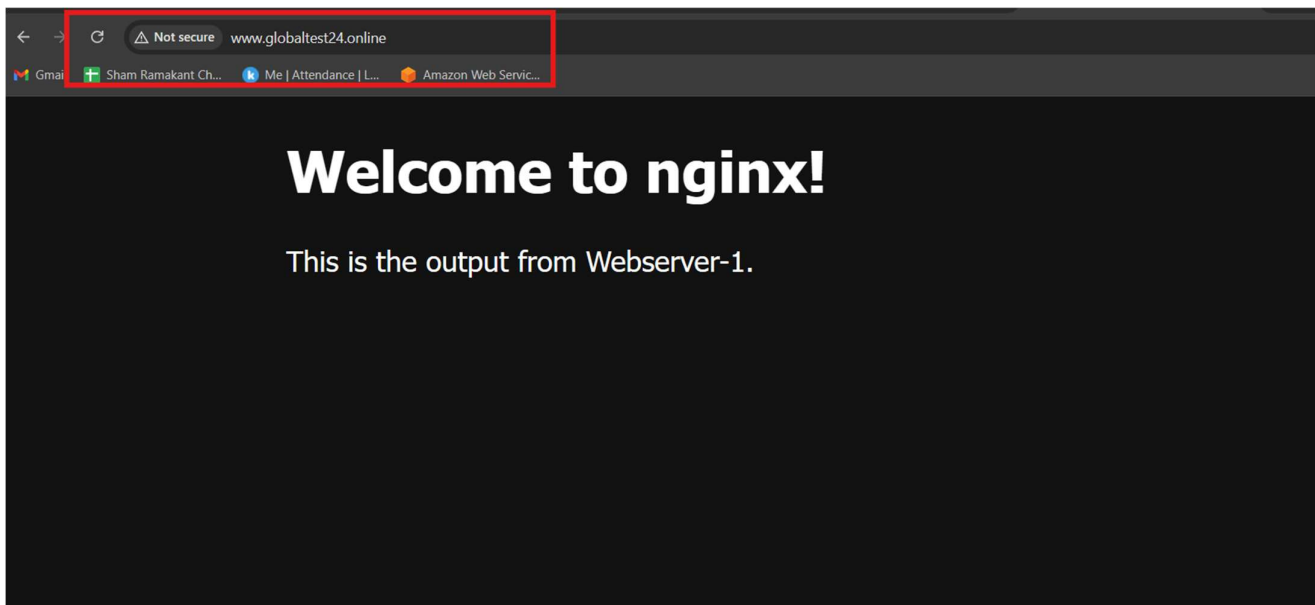
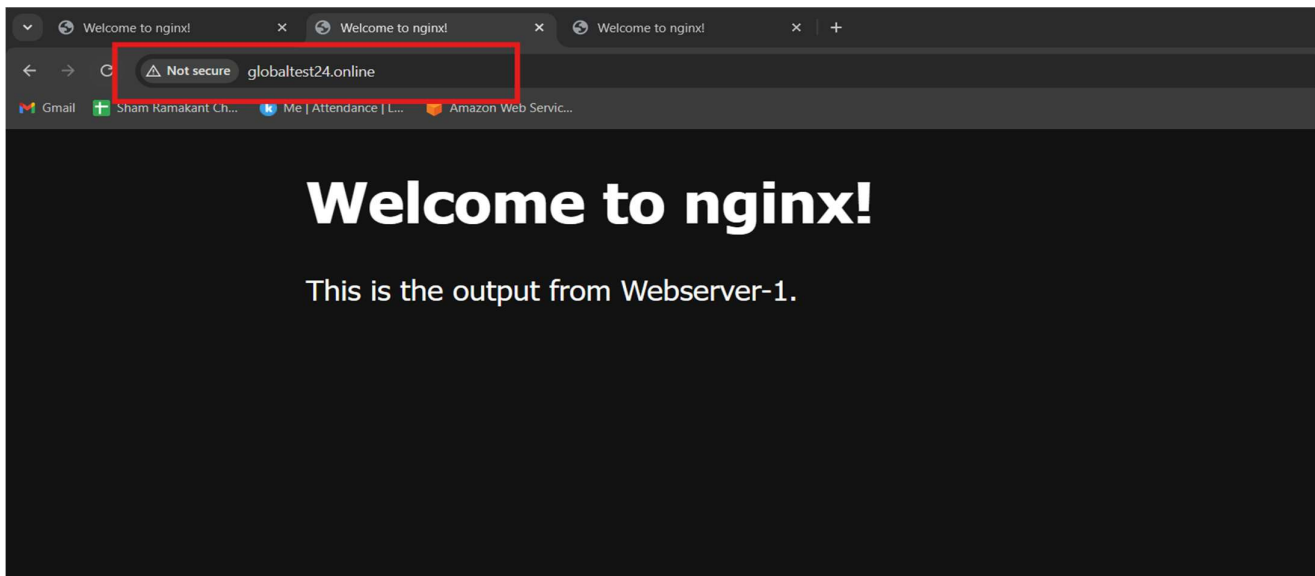
CURRENT RECORDS (1)

globaltest24.online	IN	A	IPv4 ADDRESS* 13.127.155.79
TTL (Time to Live) 4 hours			

These changes will take 4-6 hours to come into effect.

Save

Cancel



4. Mapping and Access on Server2 with Sub Damin: sham.globaltest24.online

ALL RECORDS | A AAAA MX CNAME NS TXT SRV SOA

ADD A RECORD

DOMAIN/SUB-DOMAIN	IN	A	IPv4 ADDRESS*
TTL (Time to Live) Choose TTL			

These changes will take 4-6 hours to come into effect.

+ Add Another Record

SAVE RECORD(S)

CURRENT RECORDS (2)

globaltest24.online	IN	A	13.127.155.79	▼
sham.globaltest24.online	IN	A	IPv4 ADDRESS*	▲
TTL (Time to Live) 4 hours				

These changes will take 4-6 hours to come into effect.



Save Cancel

