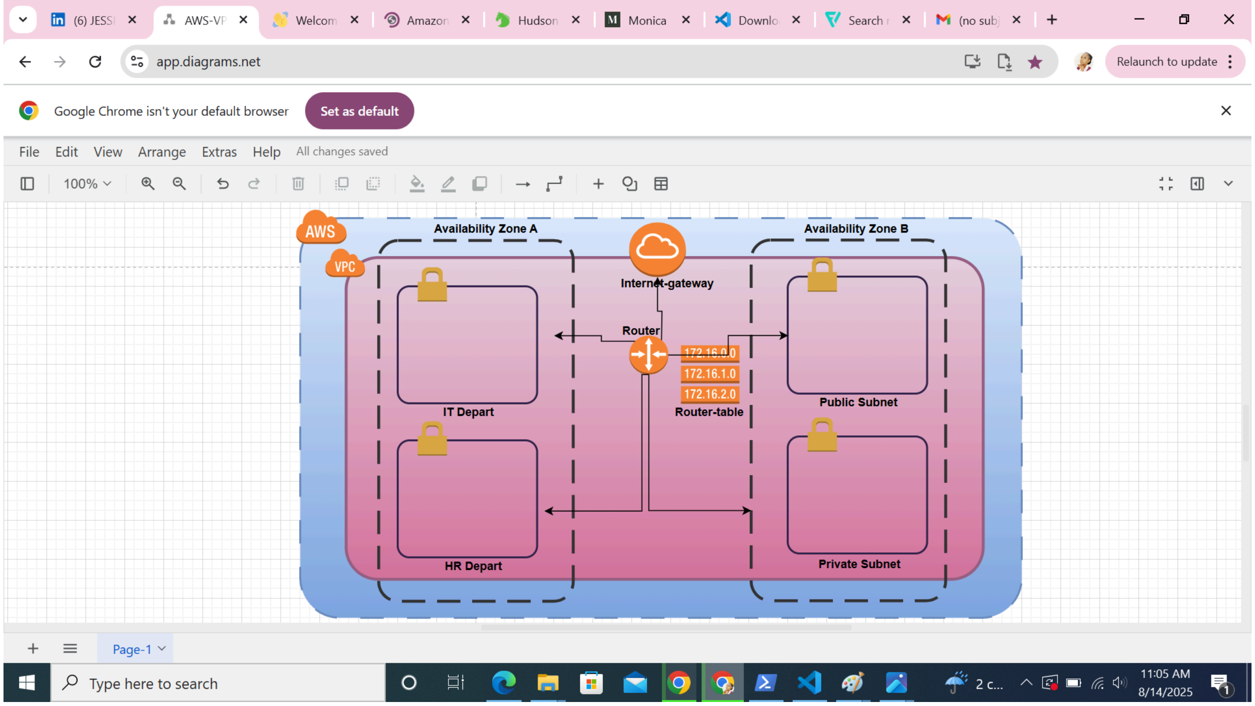
# DESIGNED AND DEPLOYED A SECURE VIRTUAL NETWORK ARCHITURE ON AWS

### PROJECT DESCRIPTION:

In this project, I successfully designed and deployed a Virtual Network Architecture on AWS (VPC), configured security with ACLs & Security Groups, and launched a live website (Medicare) inside the network.

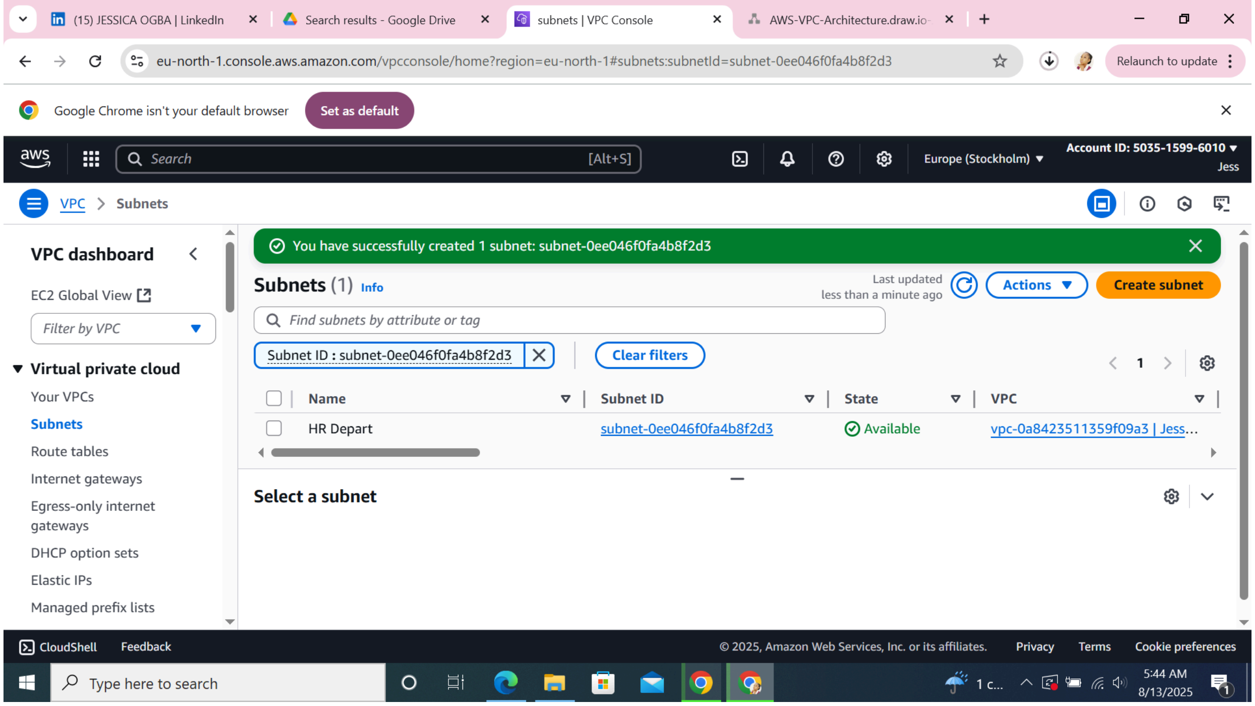
### KEY STEPS:

I Started by designing a 2-tier VPC architecture on Draw.io with the following AWS components: VPC, Public & Private Subnets, Route Tables, Internet Gateway, Security Groups, Network ACLs



## AWS Infrastructure Setup:

Next, I created the VPC, subnets, and configured routing. I launched EC2 instances (Amazon Linux) with auto-assigned public IPs, applying the required security settings for controlled access



eu-north-1.console.aws.amazon.com/vpcconsole/home?region=eu-north-1#subnets:subnetId=subnet-0ee046f0fa4b8f2d3

You have successfully created 1 subnet: subnet-0ee046f0fa4b8f2d3

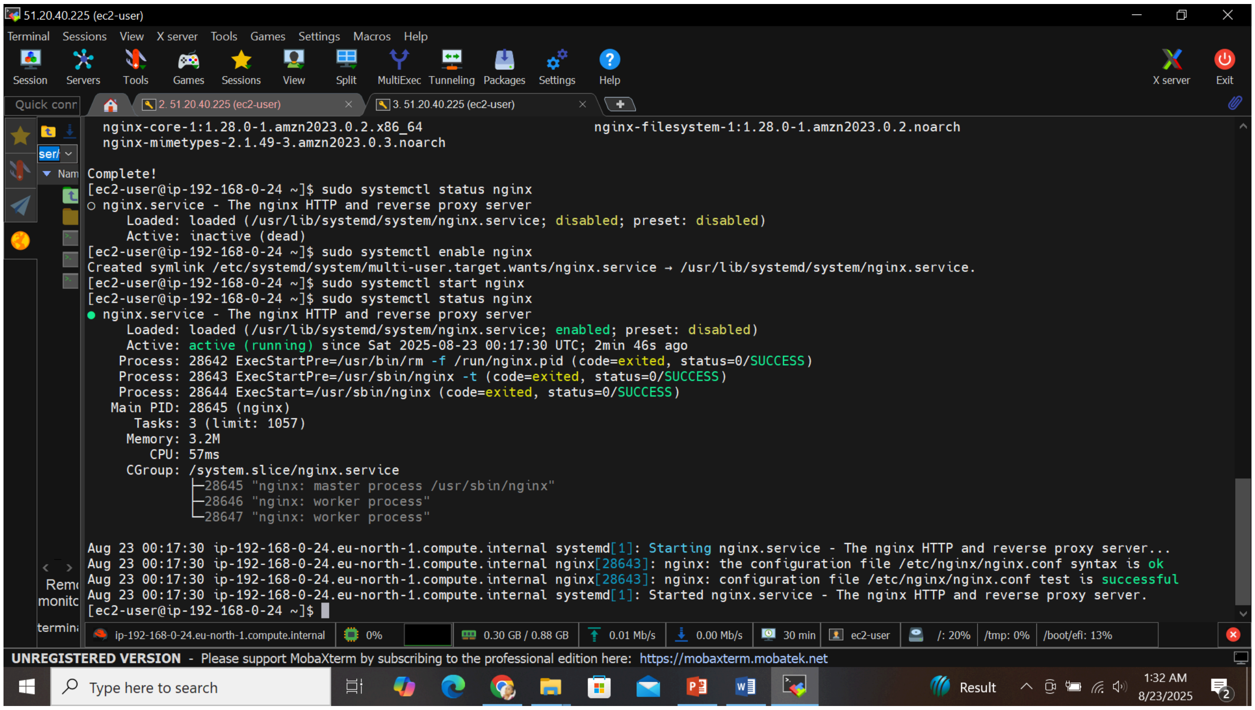
Subnets (1) Info

Subnet ID : subnet-0ee046f0fa4b8f2d3

| Name | Subnet ID | State | VPC |
|---|---|---|---|
| HR Depart | subnet-0ee046f0fa4b8f2d3 | ✓ Available | vpc-0a8423511359f09a3 \| Jess… |

## EC2 Web Server Configuration:

Using MobaXterm, I connected to the EC2 instance, installed Nginx and Git, and set up the server environment.

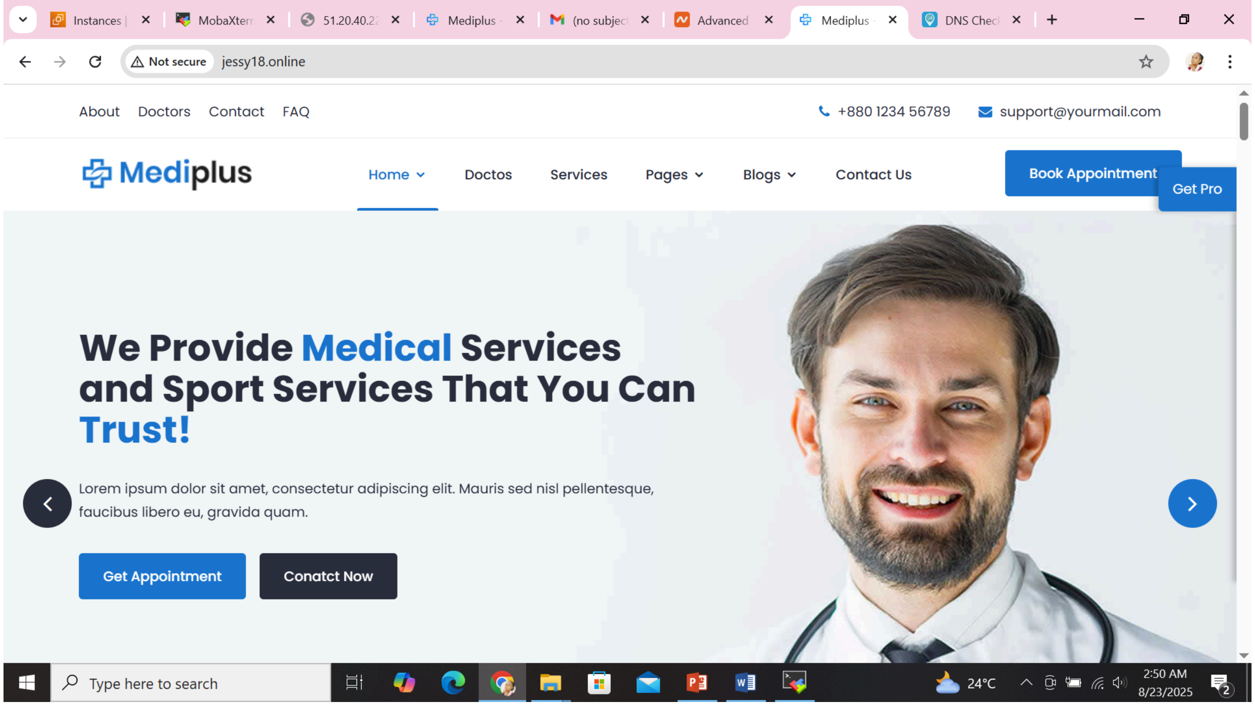

51.20.40.225 (ec2-user)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
X server Exit
Quick conr
2. 51.20.40.225 (ec2-user)
3. 51.20.40.225 (ec2-user)

```
  nginx-core-1:1.28.0-1.amzn2023.0.2.x86_64                  nginx-filesystem-1:1.28.0-1.amzn2023.0.2.noarch
  nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch

Complete!
[ec2-user@ip-192-168-0-24 ~]$ sudo systemctl status nginx
○ nginx.service - The nginx HTTP and reverse proxy server
     Loaded: loaded (/usr/lib/systemd/system/nginx.service; disabled; preset: disabled)
     Active: inactive (dead)
[ec2-user@ip-192-168-0-24 ~]$ sudo systemctl enable nginx
Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service → /usr/lib/systemd/system/nginx.service.
[ec2-user@ip-192-168-0-24 ~]$ sudo systemctl start nginx
[ec2-user@ip-192-168-0-24 ~]$ sudo systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
     Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: disabled)
     Active: active (running) since Sat 2025-08-23 00:17:30 UTC; 2min 46s ago
    Process: 28642 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
    Process: 28643 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
    Process: 28644 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
   Main PID: 28645 (nginx)
      Tasks: 3 (limit: 1057)
     Memory: 3.2M
        CPU: 57ms
     CGroup: /system.slice/nginx.service
             ├─28645 "nginx: master process /usr/sbin/nginx"
             ├─28646 "nginx: worker process"
             └─28647 "nginx: worker process"

Aug 23 00:17:30 ip-192-168-0-24.eu-north-1.compute.internal systemd[1]: Starting nginx.service - The nginx HTTP and reverse proxy server...
Aug 23 00:17:30 ip-192-168-0-24.eu-north-1.compute.internal nginx[28643]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
Aug 23 00:17:30 ip-192-168-0-24.eu-north-1.compute.internal nginx[28643]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Aug 23 00:17:30 ip-192-168-0-24.eu-north-1.compute.internal systemd[1]: Started nginx.service - The nginx HTTP and reverse proxy server.
[ec2-user@ip-192-168-0-24 ~]$
```

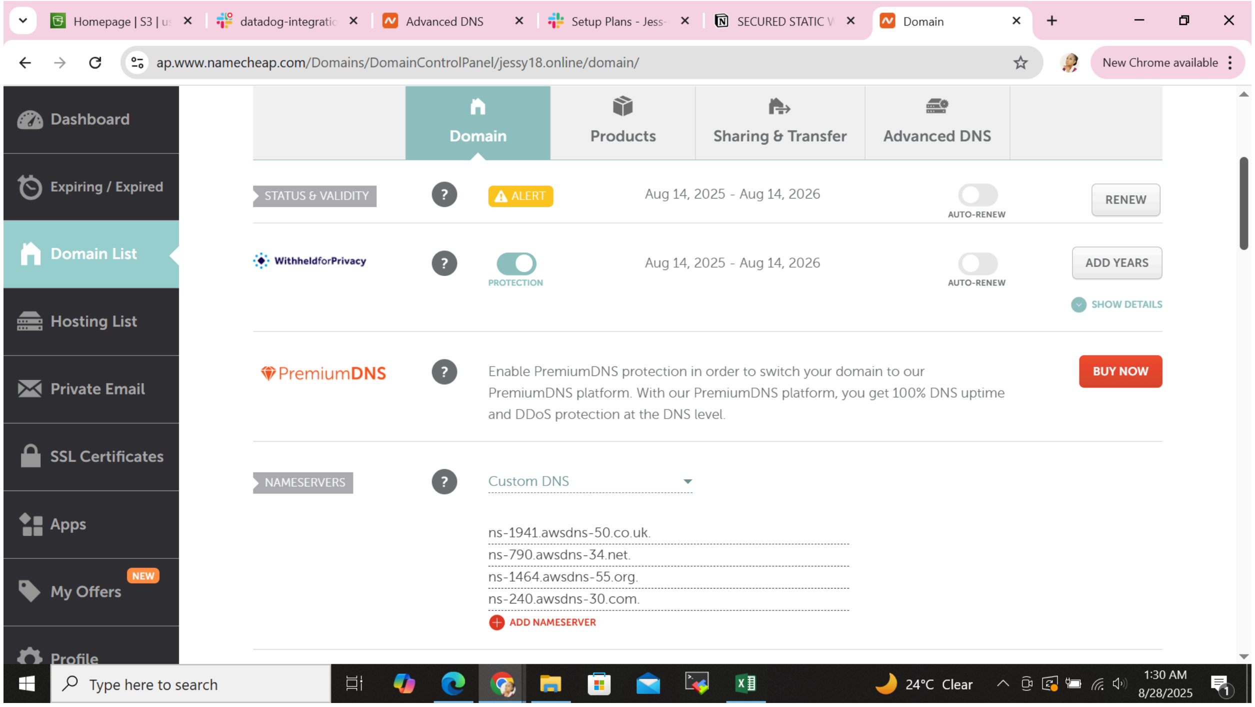
ip-192-168-0-24.eu-north-1.compute.internal
0%
0.30 GB / 0.88 GB
0.01 Mb/s
0.00 Mb/s
30 min
ec2-user
/: 20%
/tmp: 0%
/boot/efi: 13%
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net
Type here to search
Result
1:32 AM
8/23/2025

## Website Deployment:

I cloned the MediCare healthcare website from GitHub, moved the files to the Nginx web directory, and successfully hosted the site.



**Custom Domain Integration:**
Finally, I purchased a custom domain, configured DNS records, and pointed it to the server, making the website accessible publicly.



ap.www.namecheap.com/Domains/DomainControlPanel/jessy18.online/domain/

STATUS & VALIDITY: ALERT Aug 14, 2025 - Aug 14, 2026 AUTO-RENEW

WithheldforPrivacy: PROTECTION Aug 14, 2025 - Aug 14, 2026 AUTO-RENEW

PremiumDNS: Enable PremiumDNS protection in order to switch your domain to our PremiumDNS platform. With our PremiumDNS platform, you get 100% DNS uptime and DDoS protection at the DNS level.

NAMESERVERS: Custom DNS

ns-1941.awsdns-50.co.uk.
ns-790.awsdns-34.net.
ns-1464.awsdns-55.org.
ns-240.awsdns-30.com.

**Key Takeaways from this Project:**

- Designing an architecture before implementation ensures scalability and security.
- Hands-on practice with VPC, EC2, and DNS deepened my confidence in deploying real-world solutions.
- Every project strengthens my foundation and makes me more eager to contribute in professional settings.