

Nginx

Nginx is webserver application where we can

- Host your own **static website**
- Reverse proxy to a **Node.js app**
- Set up a **redirect** to another domain
- Enable **HTTPS** with Certbot

Reverse proxy

A reverse proxy server acts as an intermediary, sitting in front of one or more web servers and receiving client requests. It then forwards these requests to the appropriate backend servers, effectively hiding the internal server structure from the outside world.

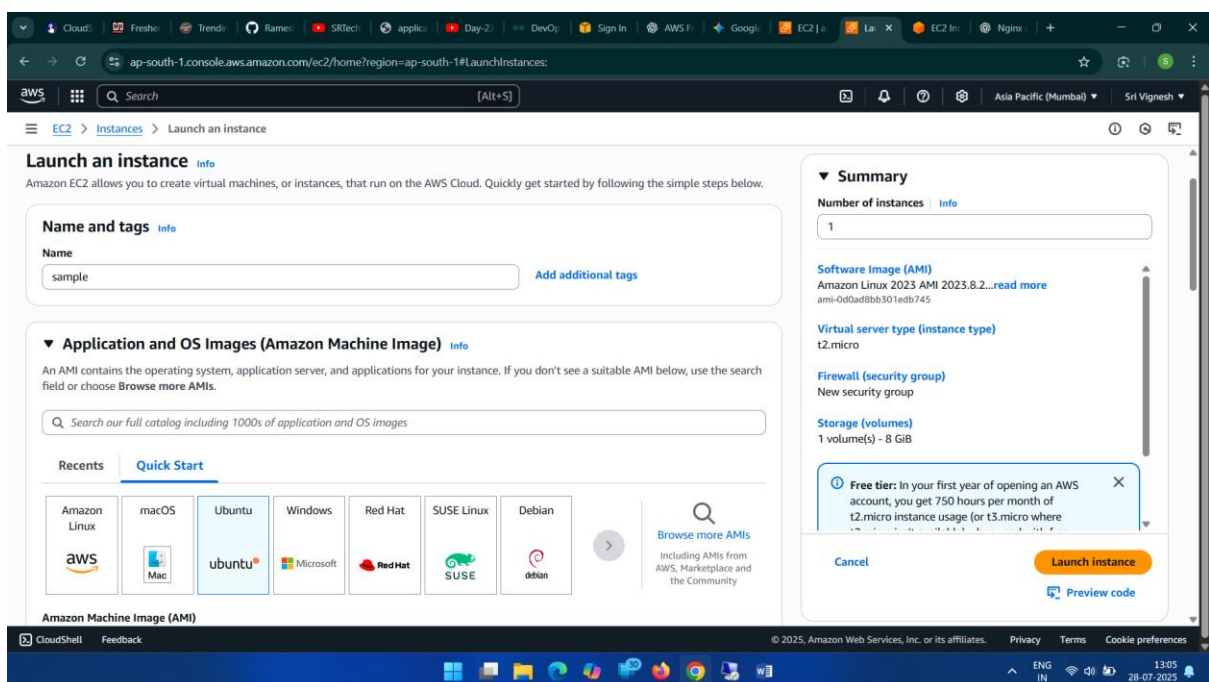
Reverse proxy vs Load Balancing

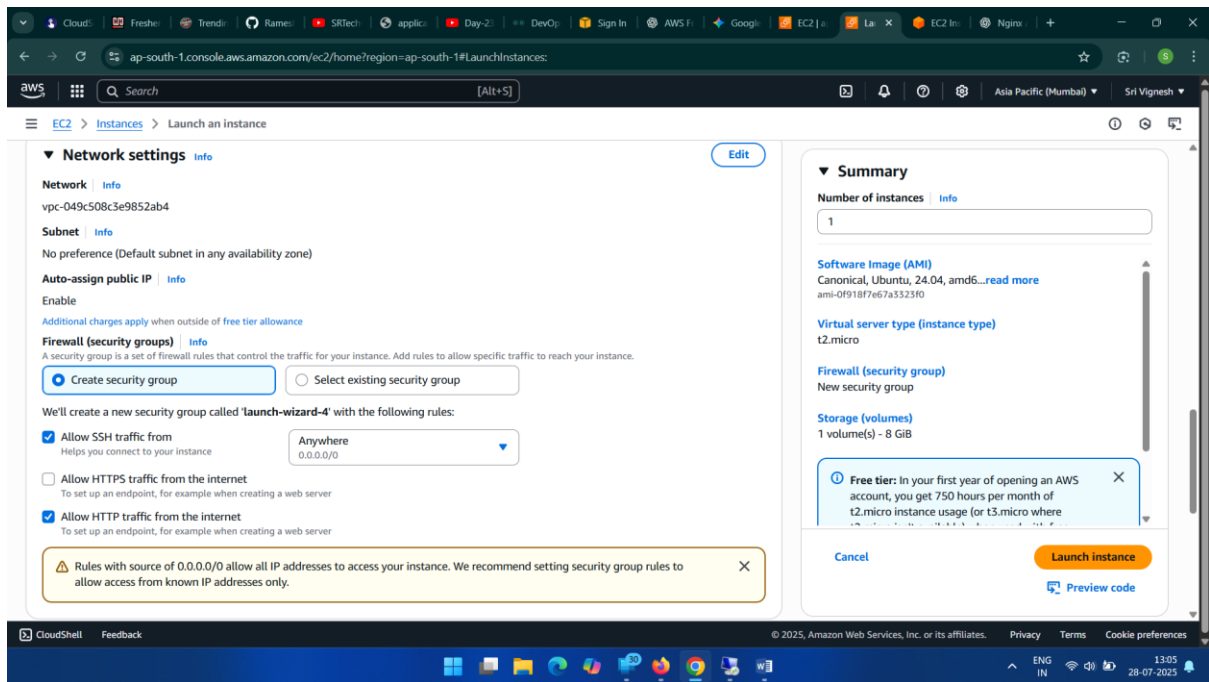
A reverse proxy acts as an intermediary, handling client requests and forwarding them to backend servers, often focusing on security, caching, and SSL termination.

Load balancers, on the other hand, distribute incoming traffic across multiple servers to prevent any single server from being overwhelmed, ensuring high availability and scalability

Difference between Nginx and Apache

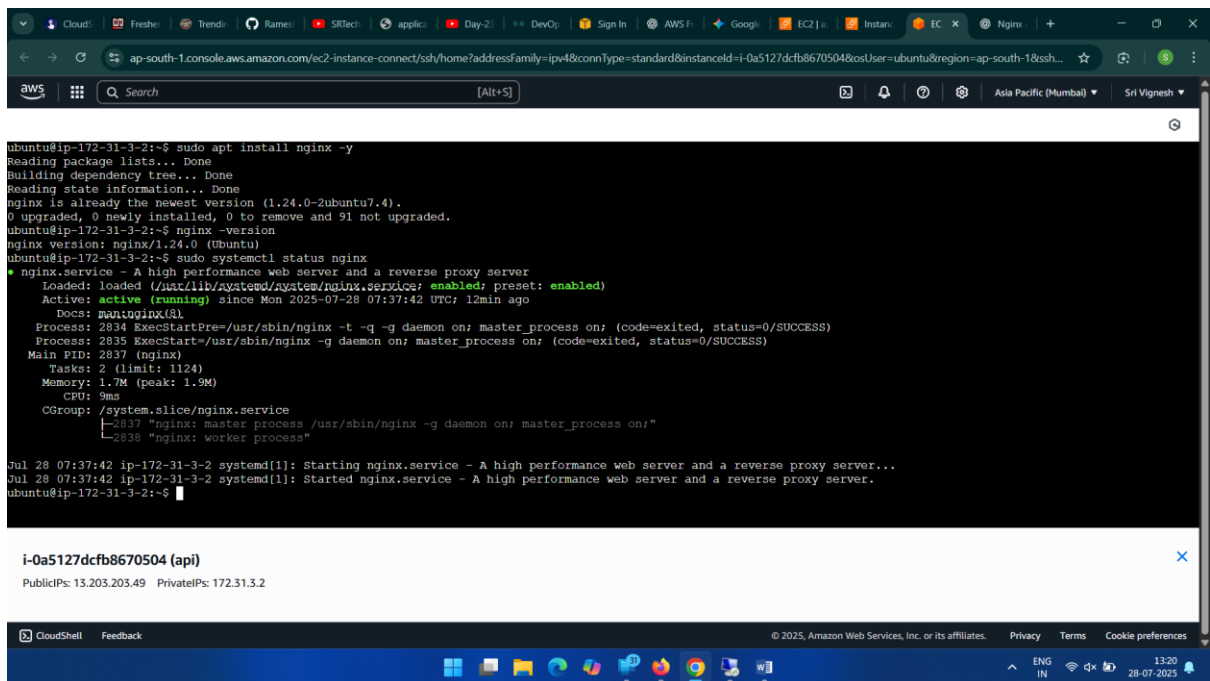
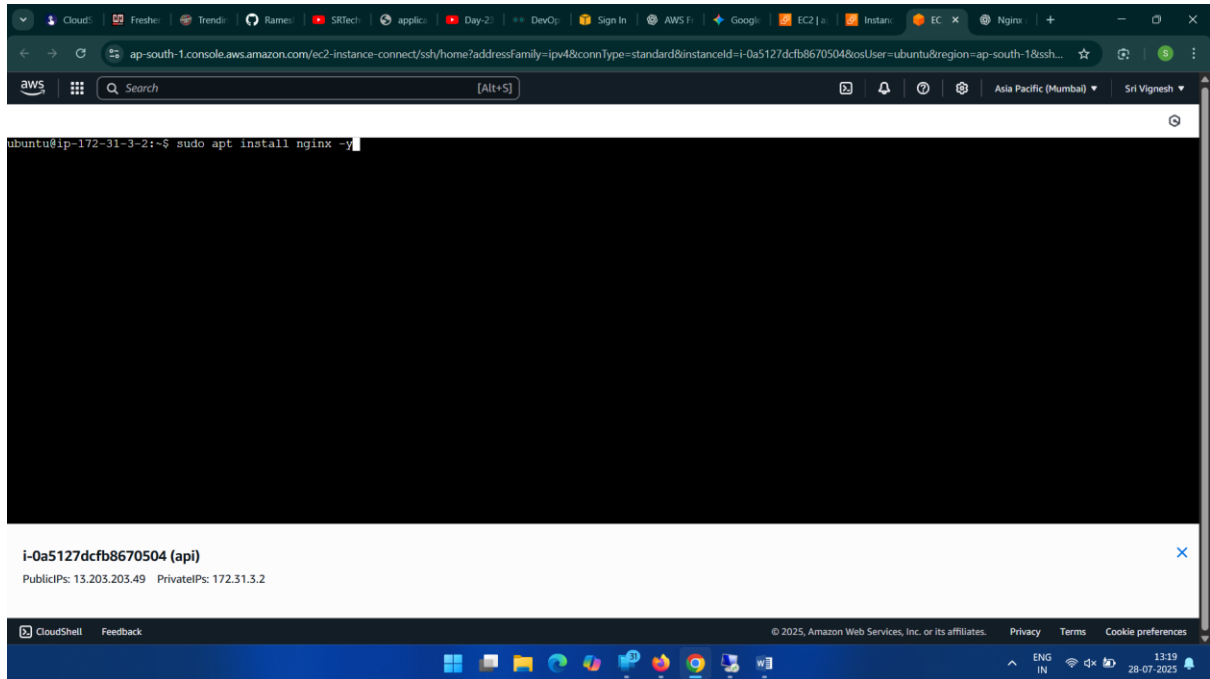
Nginx is known for handling high traffic efficiently, while Apache is more versatile but can be slower under heavy loads.

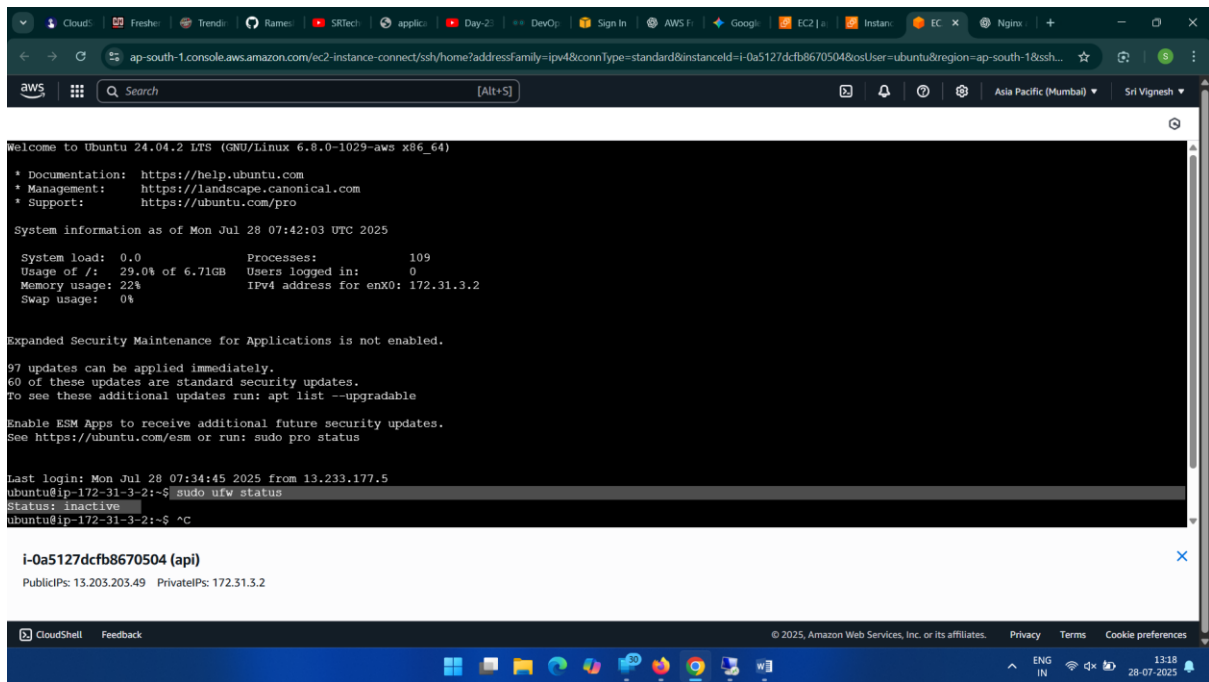




Connect to the instance

- `sudo apt update`
- `sudo apt install nginx -y`
- `nginx -version`
- `sudo systemctl status nginx`
- `sudo systemctl start nginx`
- `sudo systemctl enable nginx`
- `sudo systemctl stop nginx`
- `sudo apt remove nginx -y`
- `sudo ufw status`
 1. If it says: `inactive` → Good, nothing blocking.
 2. If it says: `active` → Could be blocking port 80.
- Link to reach the running webserver : instance public Ip
 1. <http://65.2.71.137> (example ip)
 2. `curl http://65.2.71.137`
 3. `ping 65.2.71.137`





Now, the nginx is running and not blocking anything from external world. Then go to the public Ip, in this case “ <http://13.203.203.49/>” -> ensure http but not https (not works) reason: security group allowing ssh and http request only but not https and nginx default http port is 80. With above link the webpage is reachable

