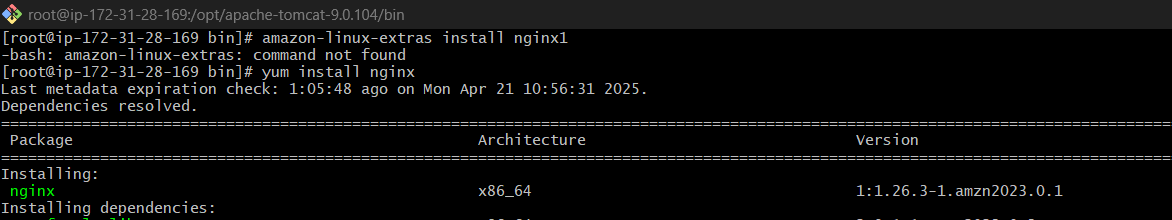
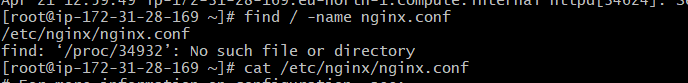
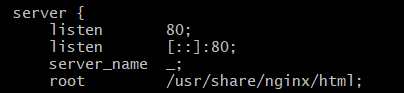
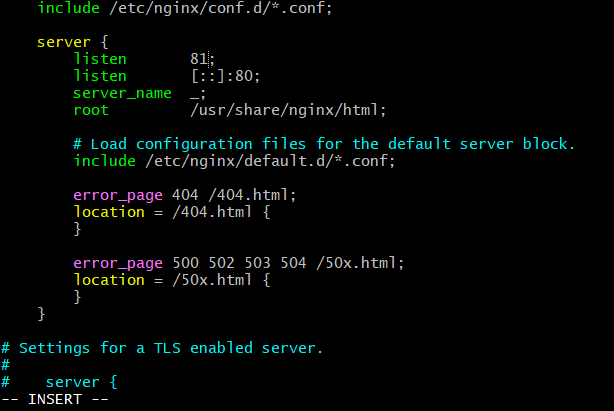
**App Server & Web Server Tasks**

*1) Install nginx and run nginx on port number 81*

* Install nginx
* Find where the nginx.conf file is and read the file to see its default port



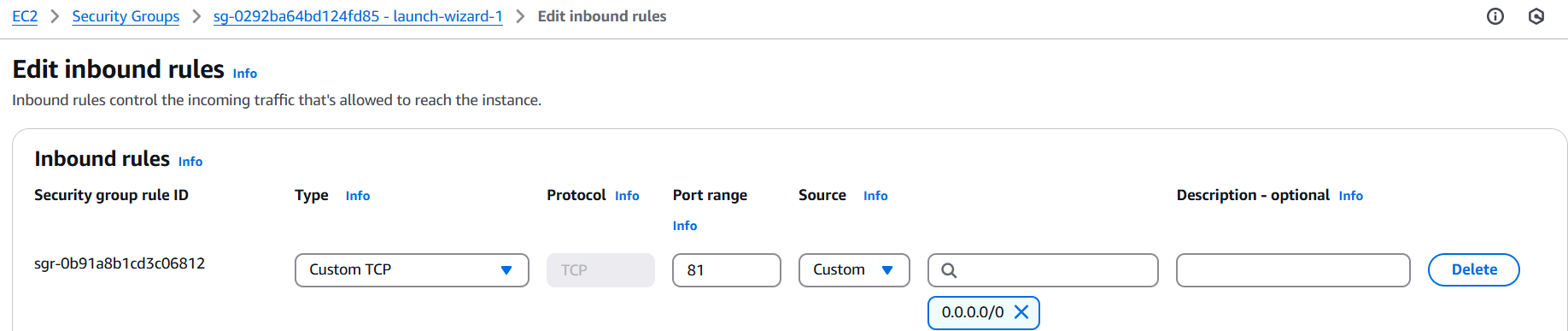
* Open the file to edit and change - listen 80 to listen 81, save and quit



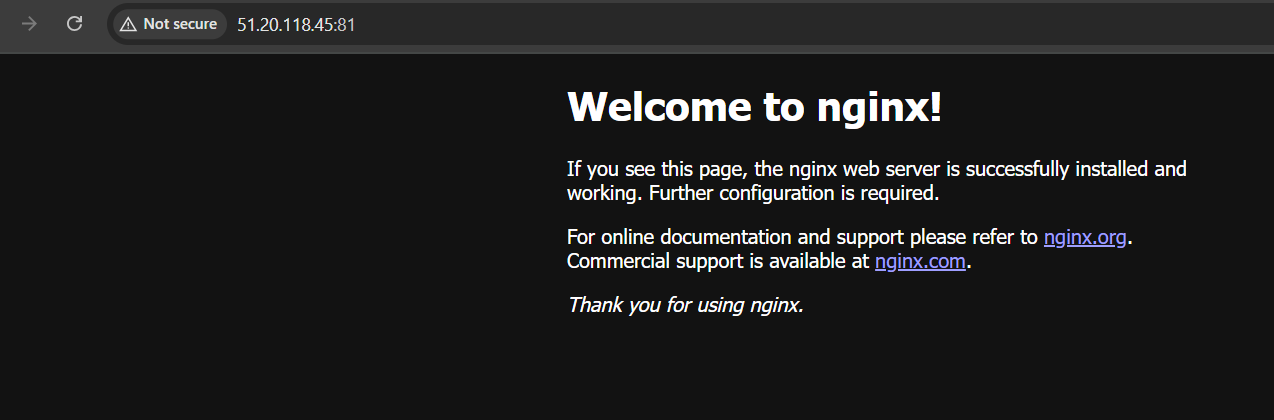
* Restart Nginx to apply changes done in config. file

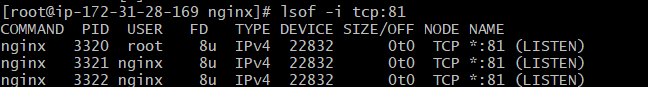


* Add new inbound rules for port 81 in the EC2 instance under security→security groups



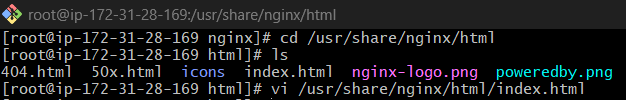
* Check if it is running on port 81



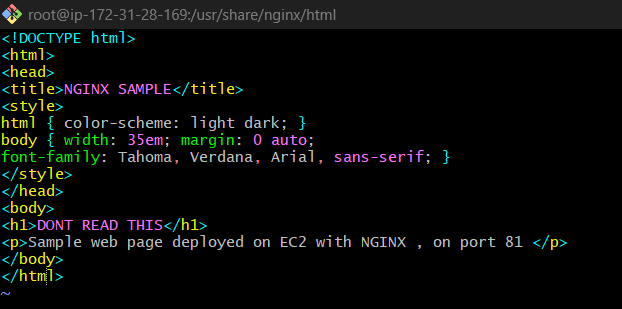


*2) Deploy a sample index.html file on nginx*

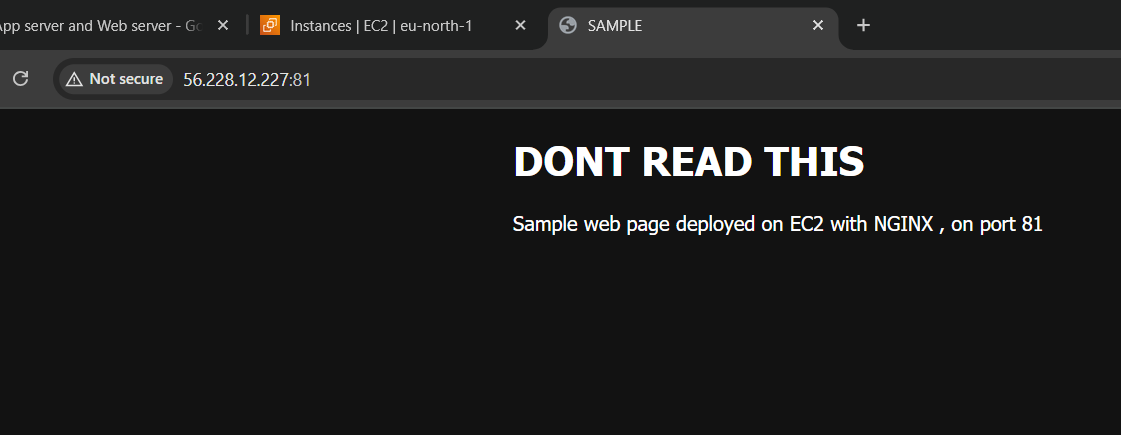
* Go to default deploy directory of nginx, where html files are located



* Open the index.html file then edit, save and quit.

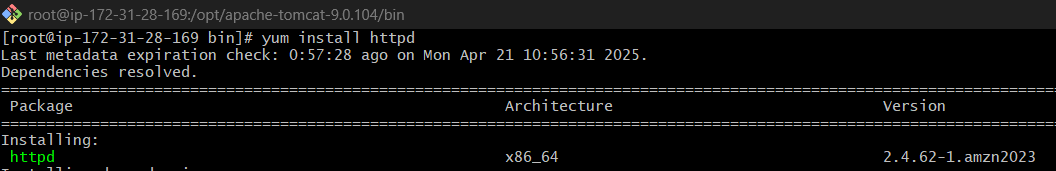


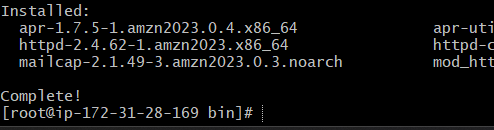
* Reload the web page and it should change.



*3) Install Apache and run Apache on port number 82*

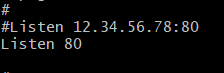
* Install Apache

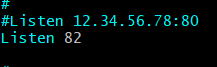


******

* Go to the default directory where the httpd.conf file is and read the file to see its default port. Open conf file , change listen 80 to 82 , save and quit.







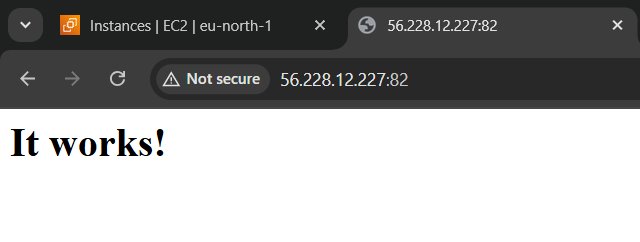
* Restart httpd to apply changes made

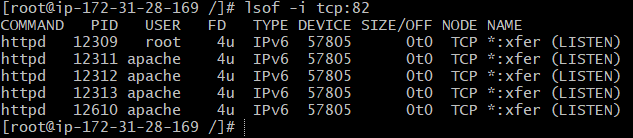


* Add new inbound rules for port 82 in the EC2 instance under security→security groups

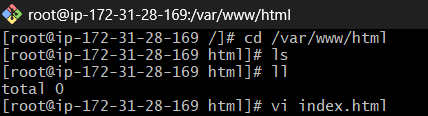


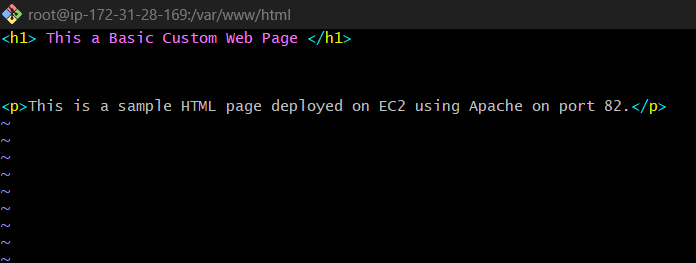
* Check if it is running on port 82



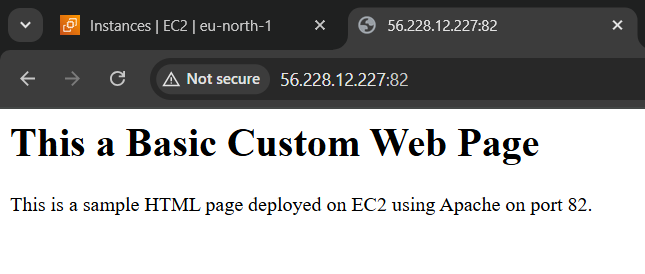


*4) Deploy a sample index.html file on Apache*

* Go to Apache’s default root directory and create a index.html. Then open it, edit, save and quit.

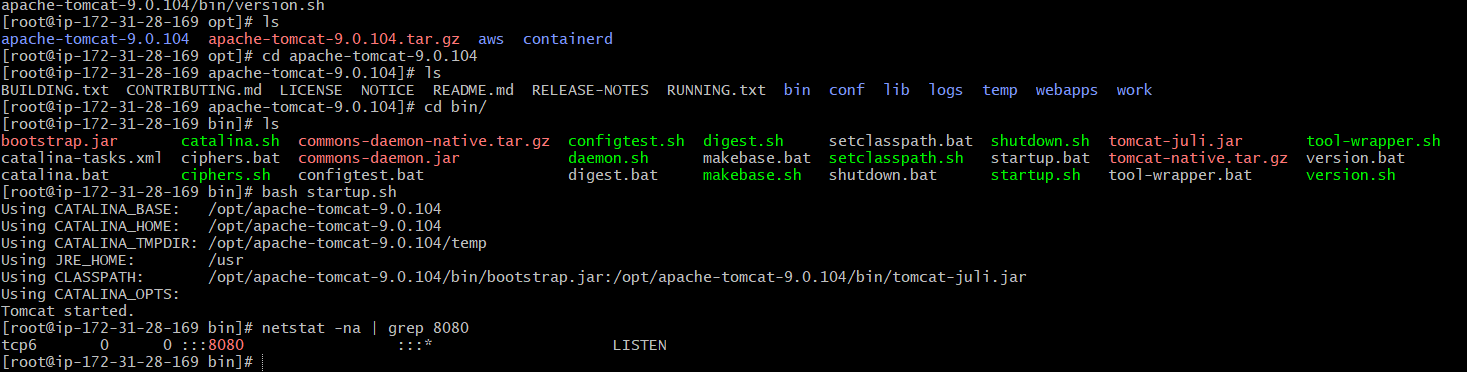


* Reload or open a new tab and test in the browser



*5) Install Apache tomcat on port number 8082*

* Install Tomcat and run it



* go the conf directory where tomcat’s server.xml file is , open and edit it. Change port ‘8080’ to ‘8082’. save and close the file.





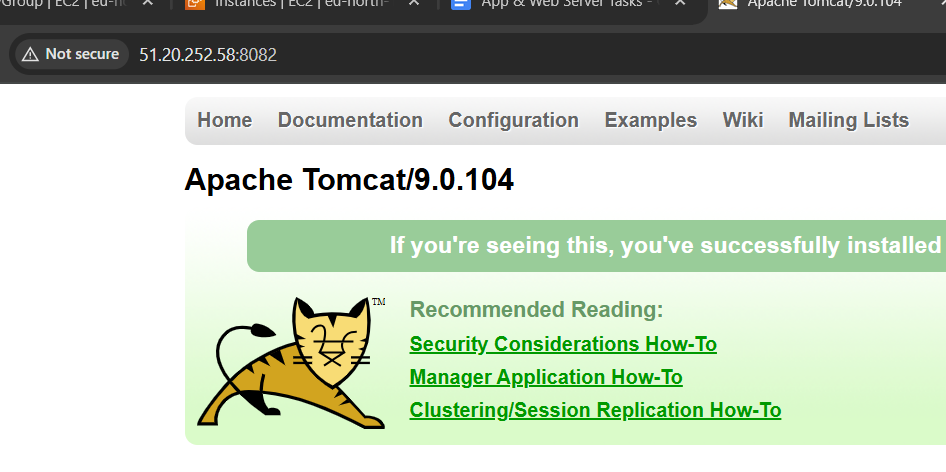


* Add new inbound rules for port 8082 in the EC2 instance under security→security groups



* Check if the port has changed to 8082 through commands or by going to the browser





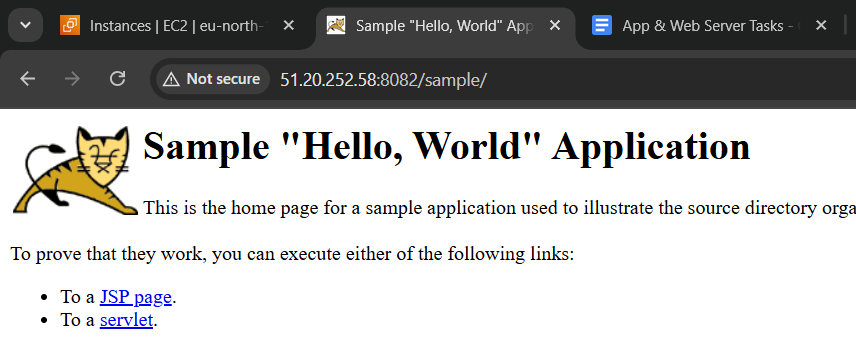
6) *Deploy a sample app on webapps*

* Go to tomcat apache’s website and get the link for sample.war , and use wget command to download the file.
* Make sure to place the . war file only under tomcat/webapps for it to work.



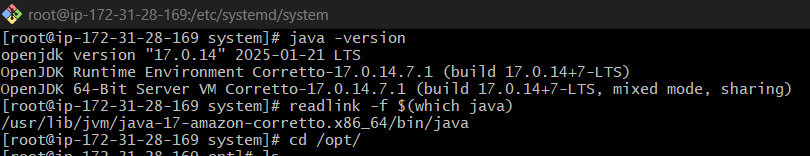


* Now go to the browser and check.

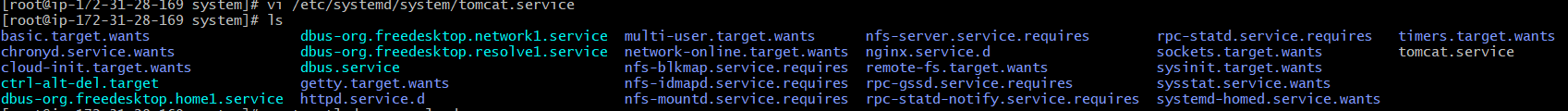


*7) Create a tomcat.service file for tomcat.*

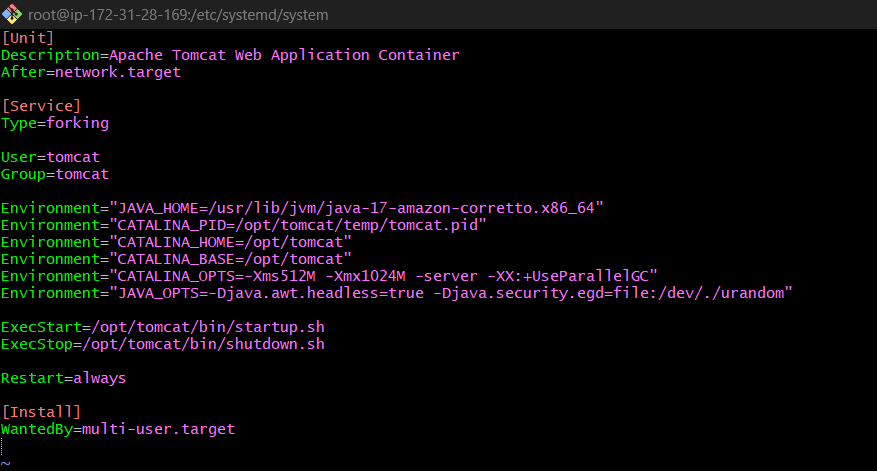
* check java version and path



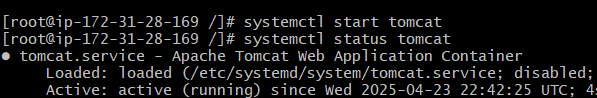
* create a tomcat.service file in /etc/systemd/system.

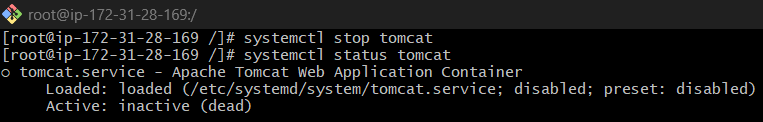


* open and configure it with the right instruction and save&quit.



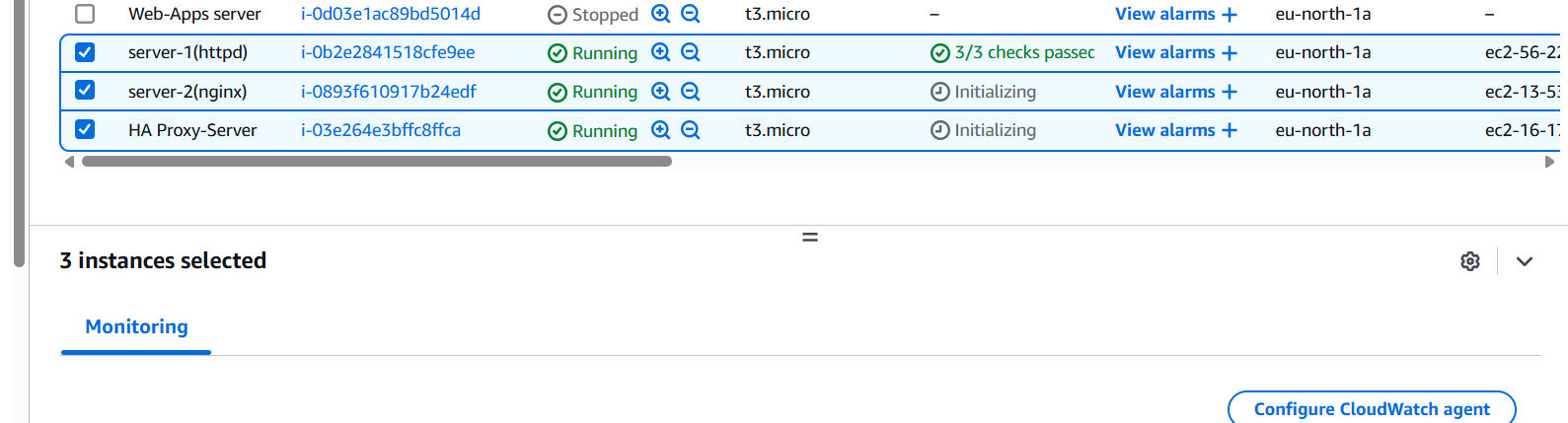
* Systemctl status tomcat,Systemctl start tomcat and Systemctl stop tomcat are now available to use.



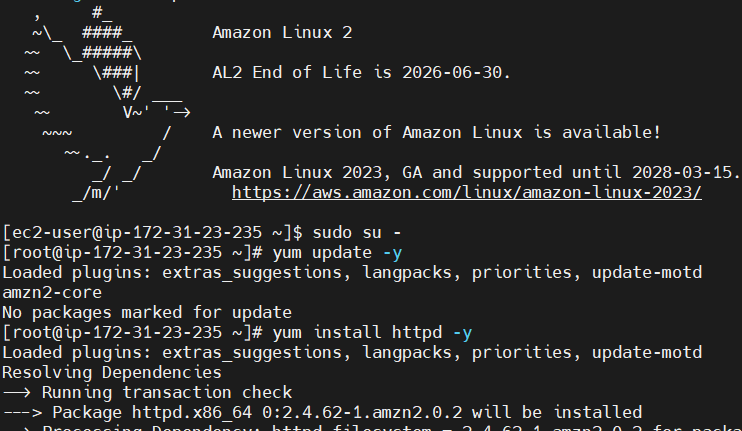


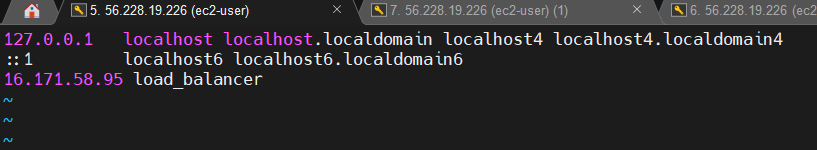
*8) Configure HA Proxy server*

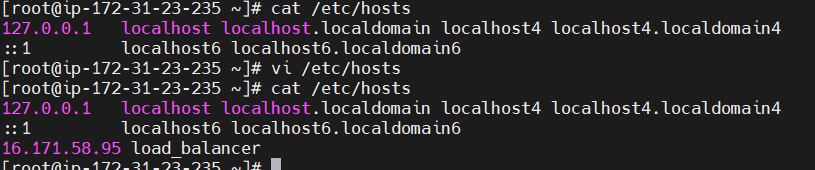
* Launch 3 new Ec2 instance and name them for ease



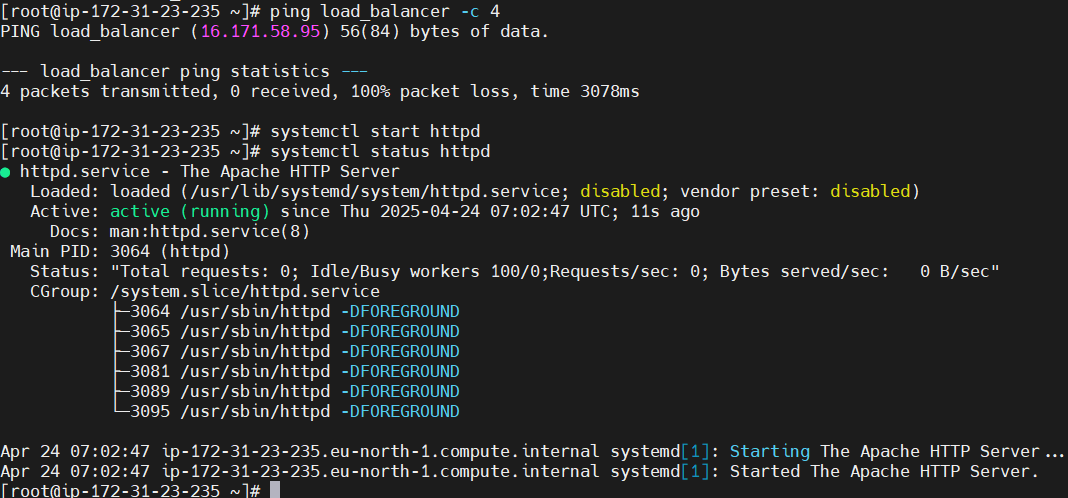
* connect to first server and install httpd

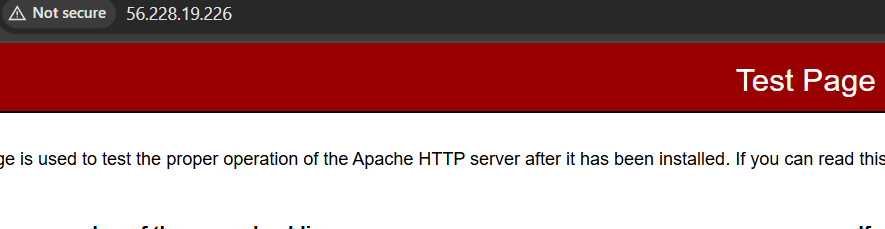


* go to /etc/hosts and edit it. add <proxy server public ip> load\_balancer



* run ping load\_balancer -c 4 and run it. Check in the browser to verify



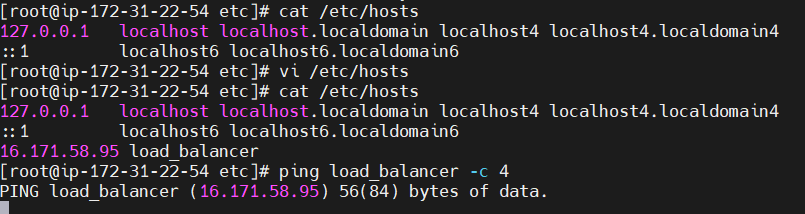


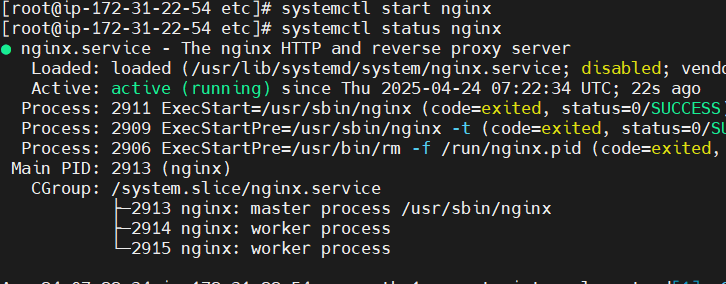
**SERVER-2**

* Repeat the same process on server two, only difference is we are using nginx . so we install nginx and the rest is same

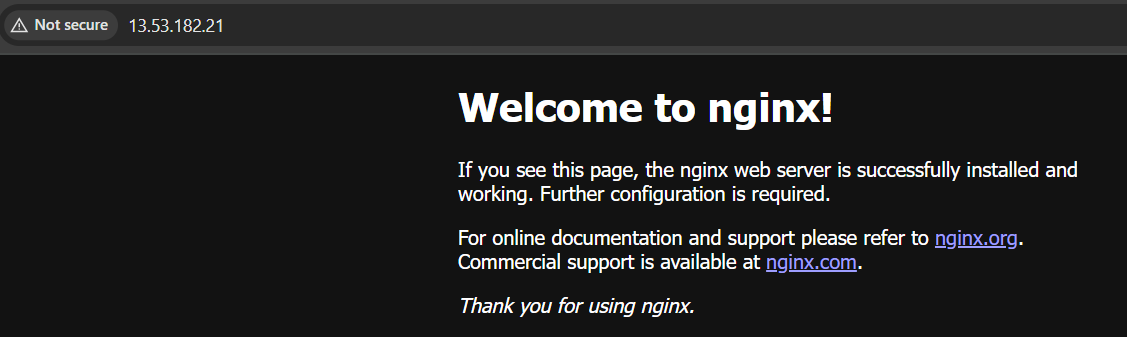


* Do the same to etc/hosts/ as for server 1 and run it



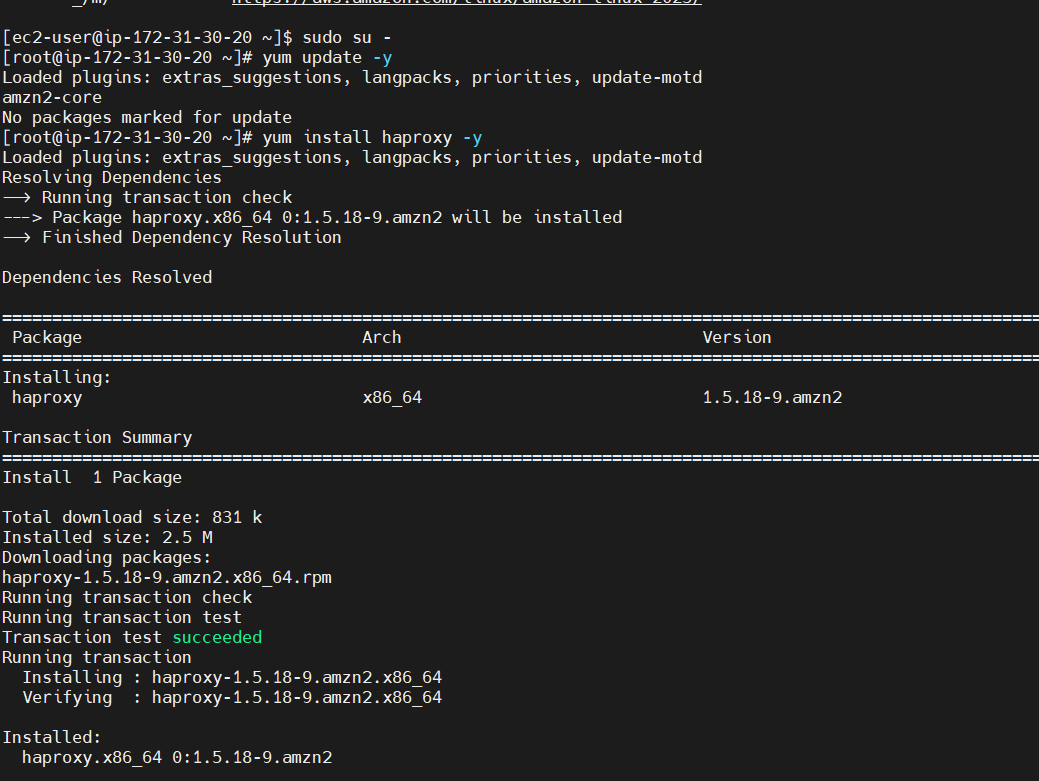


* check if it works

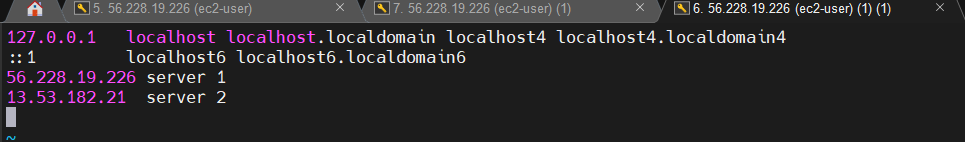


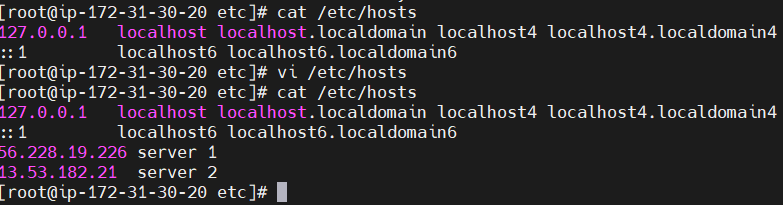
**SERVER-3**

* Install haproxy

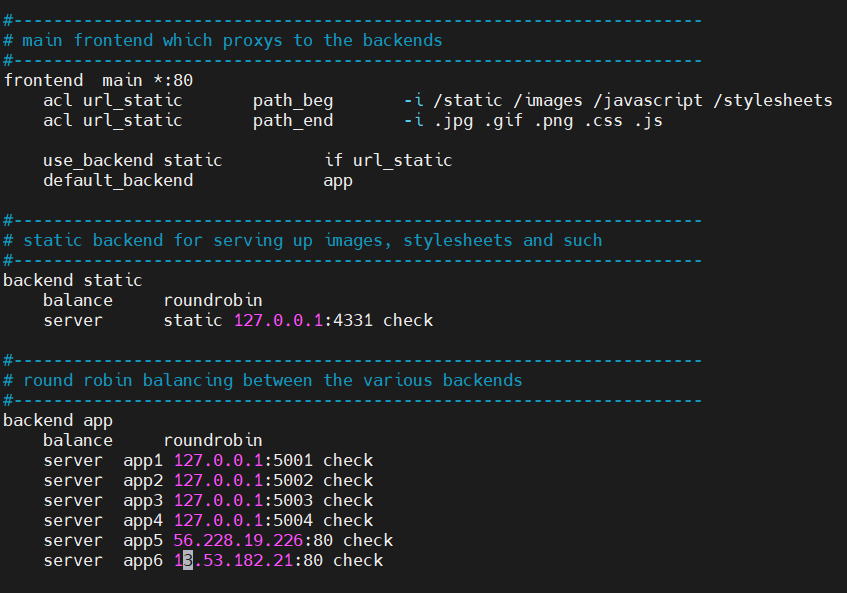


* do vi /etc/hosts and add server 1 and 2’s public IP’s

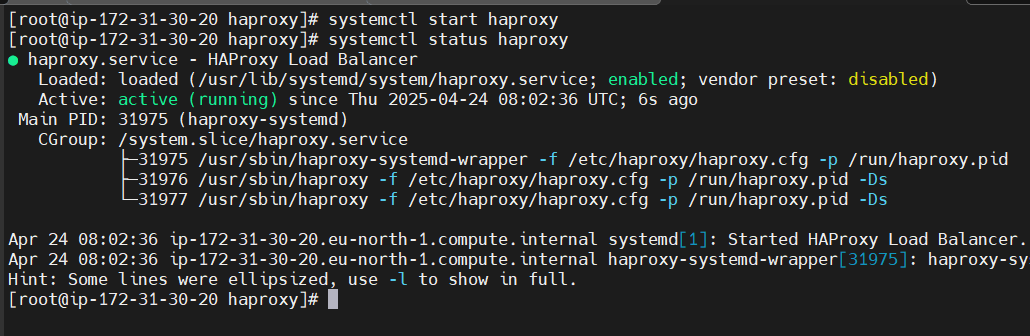




* do vi /etc/haproxy/haproxy.cfg add server 1 and 2’s public IP’s in the backend app section and change frontend main to 80.



* run it



* Now check using the “curl http://<HAProxy\_Public\_IP>” or on open on browser and keep refreshing to see it change

