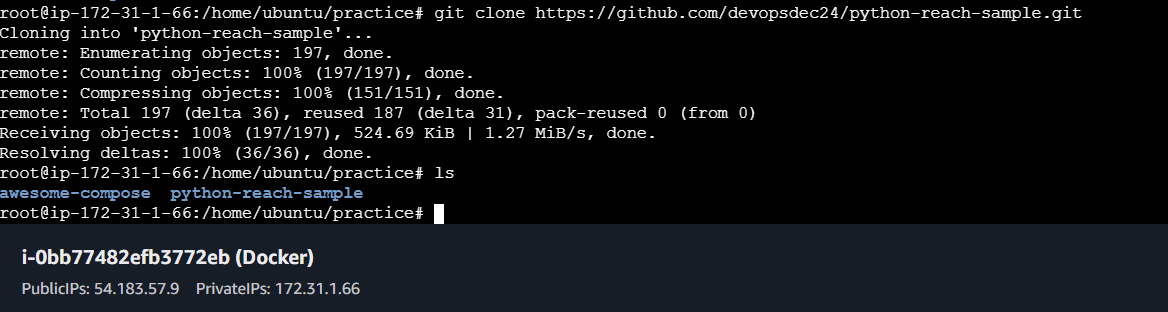
**Create the Static website from git repo mentioned below and host it using ROUTE 53**

**GIT: <https://github.com/devopsdec24/python-reach-sample.git>**

STEP1:

Clone the mentioned git to the docker machine:

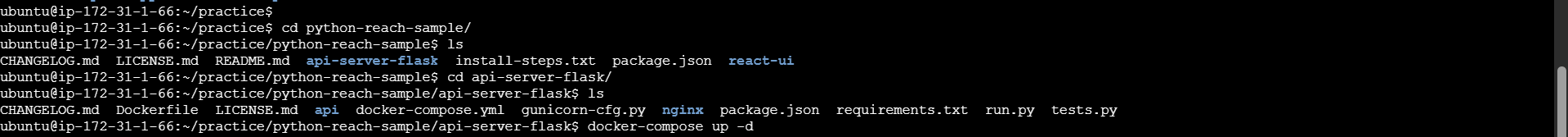
git clone <https://github.com/devopsdec24/python-reach-sample.git>

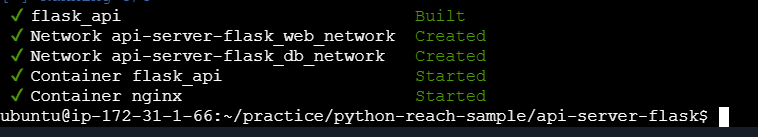


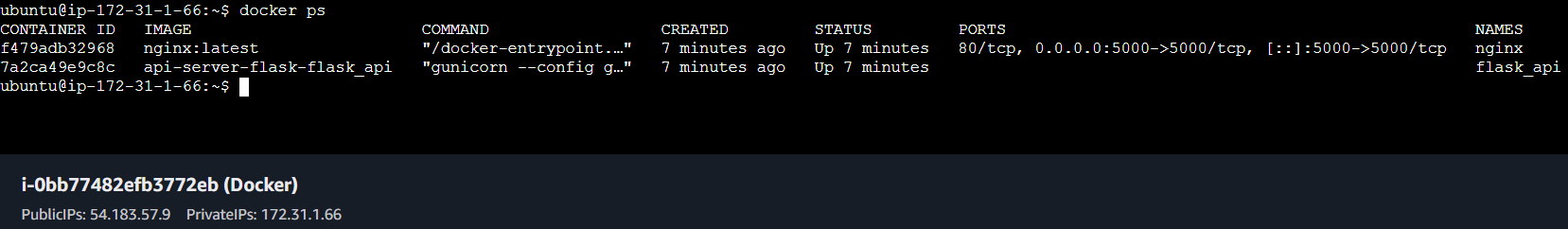
STEP 2:

Move to api-server-flask directory and execute docker-compose to create the docker container for nginx and

api-server-flask-flask\_api







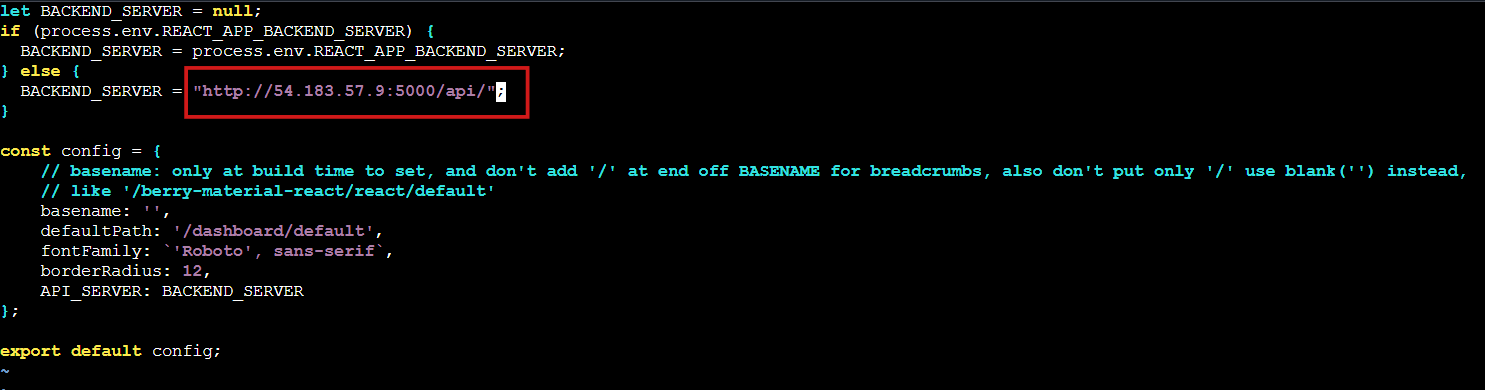
STEP 3:

Move to src and change the IP to your local machine IP as it is a static IP used by docker

machine:

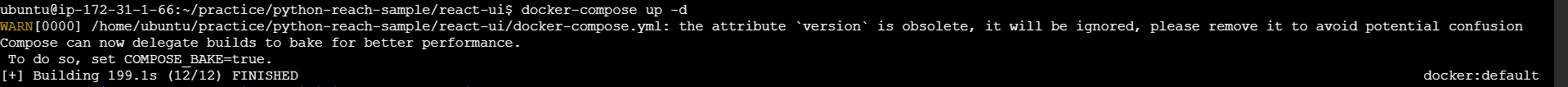
pwd: /home/ubuntu/practice/python-reach-sample/react-ui/src

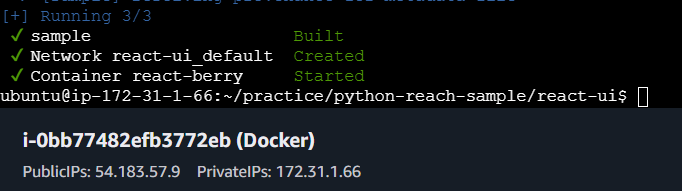
vi config.js



STEP 4:

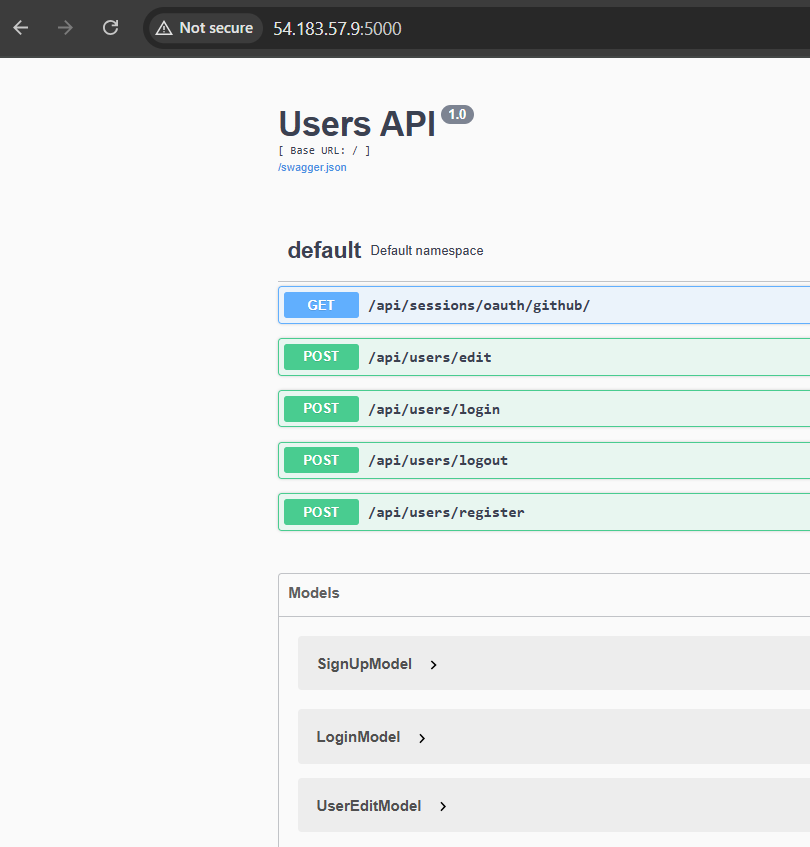
Build the docker ui machine using the below command:





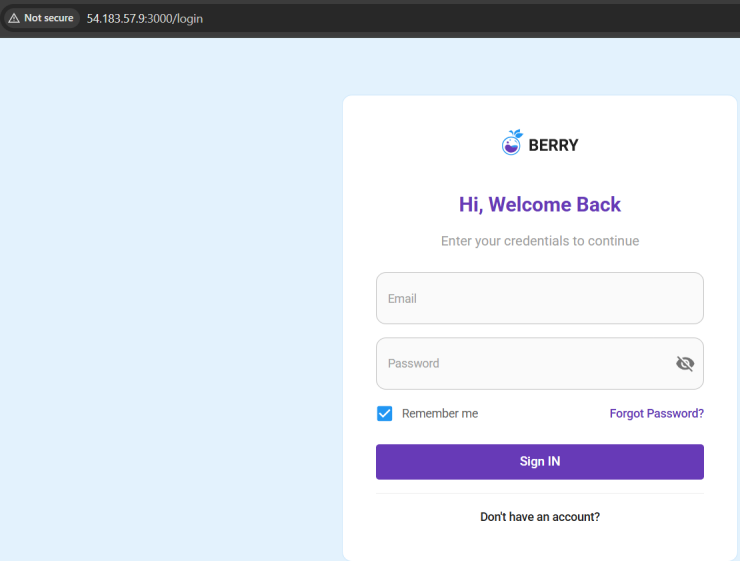
STEP 5:

Use IP with port number to check whether the API is working or not:



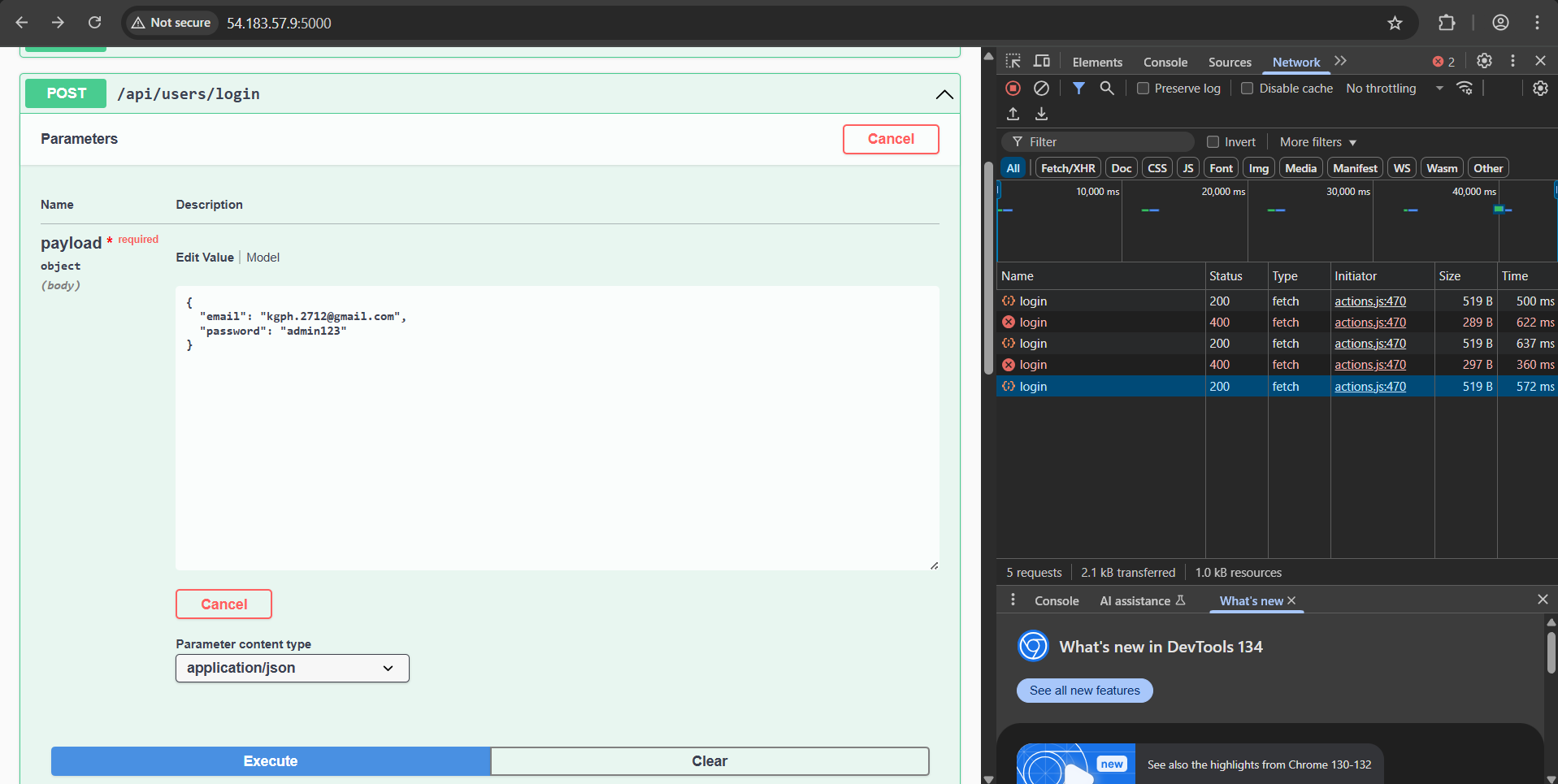
STEP 6:

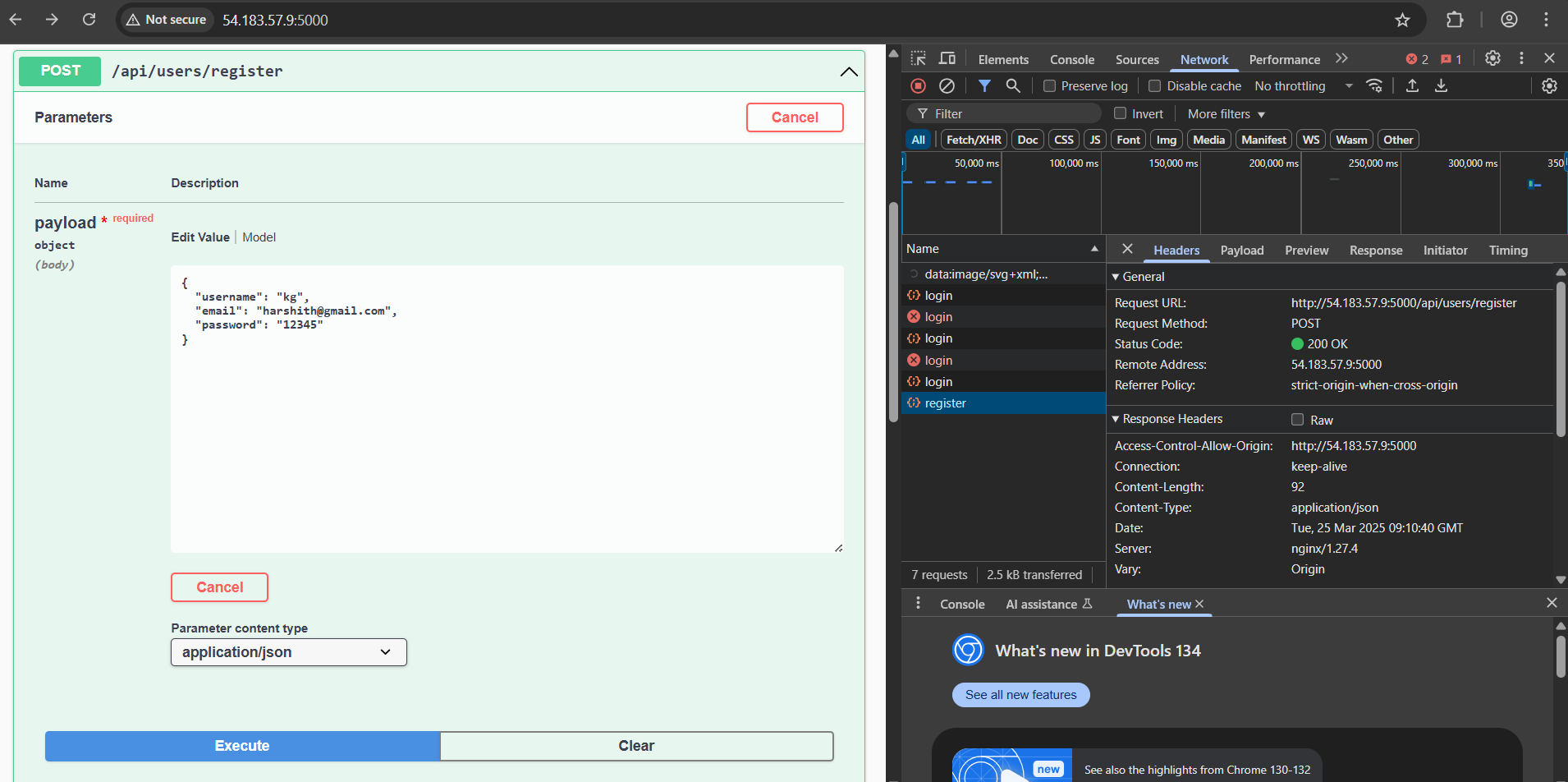
Check whether the UI is working or not using 3000 port:

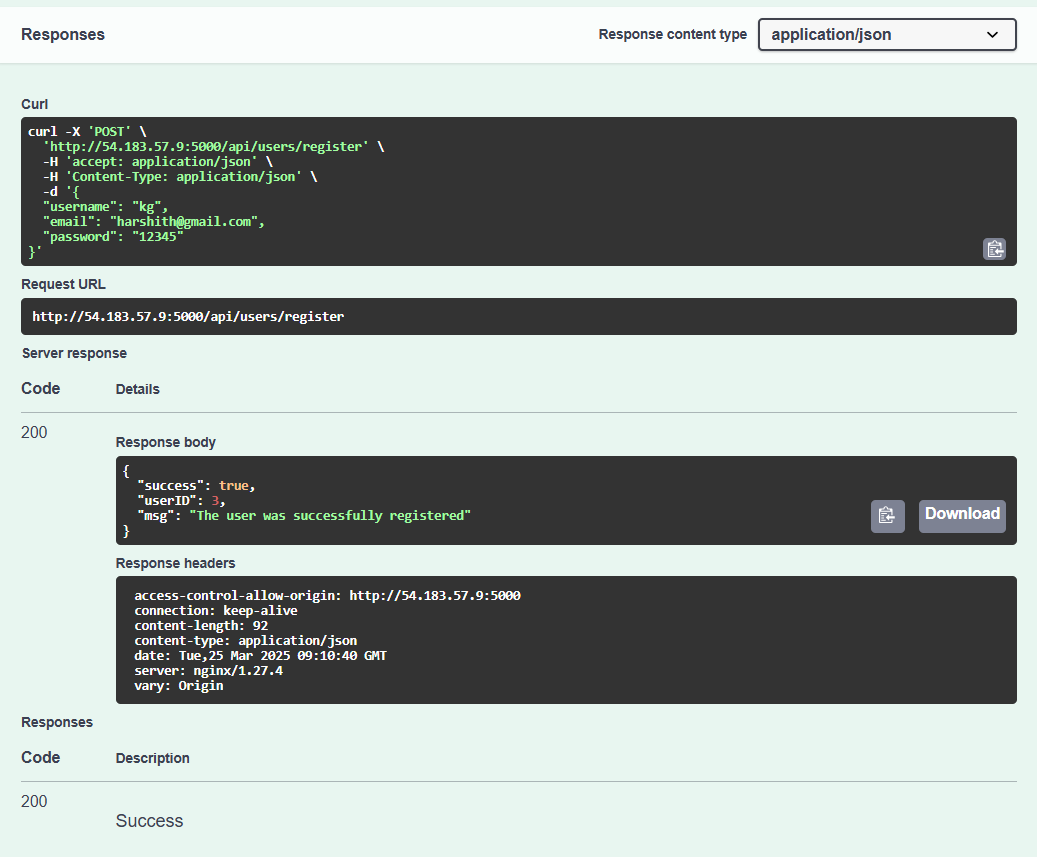


STEP 7:

Use response API and send request:



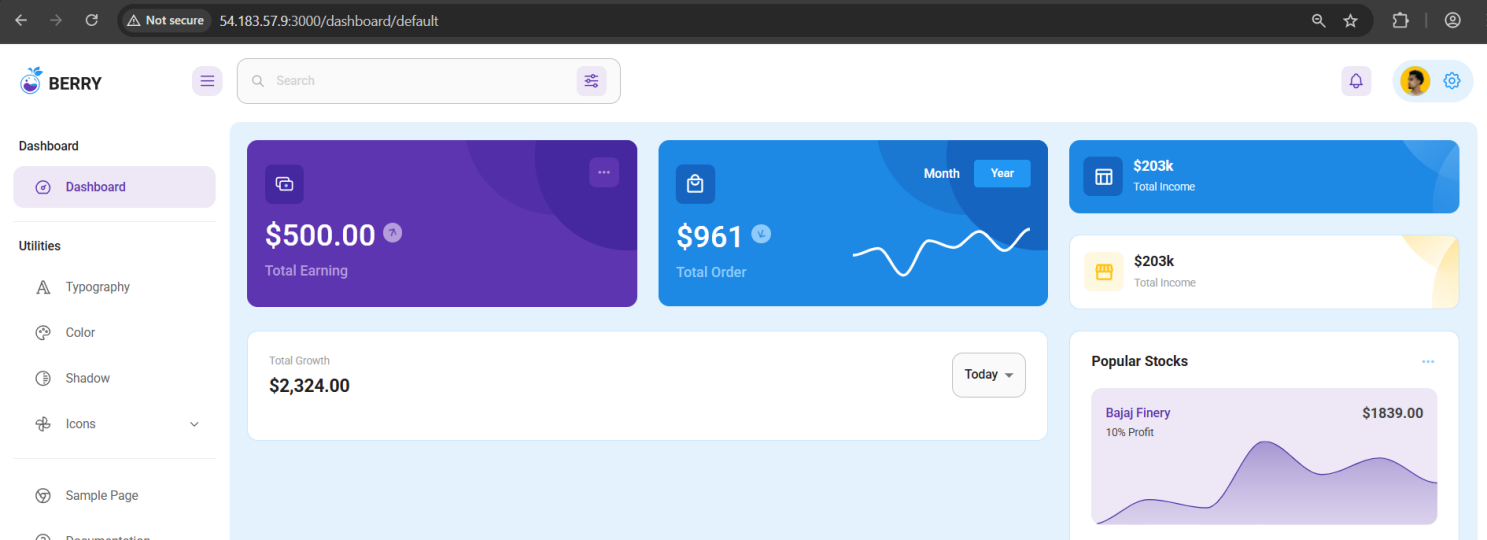




STEP 8:

Try to login with the profile which you used to register and it should take you to the

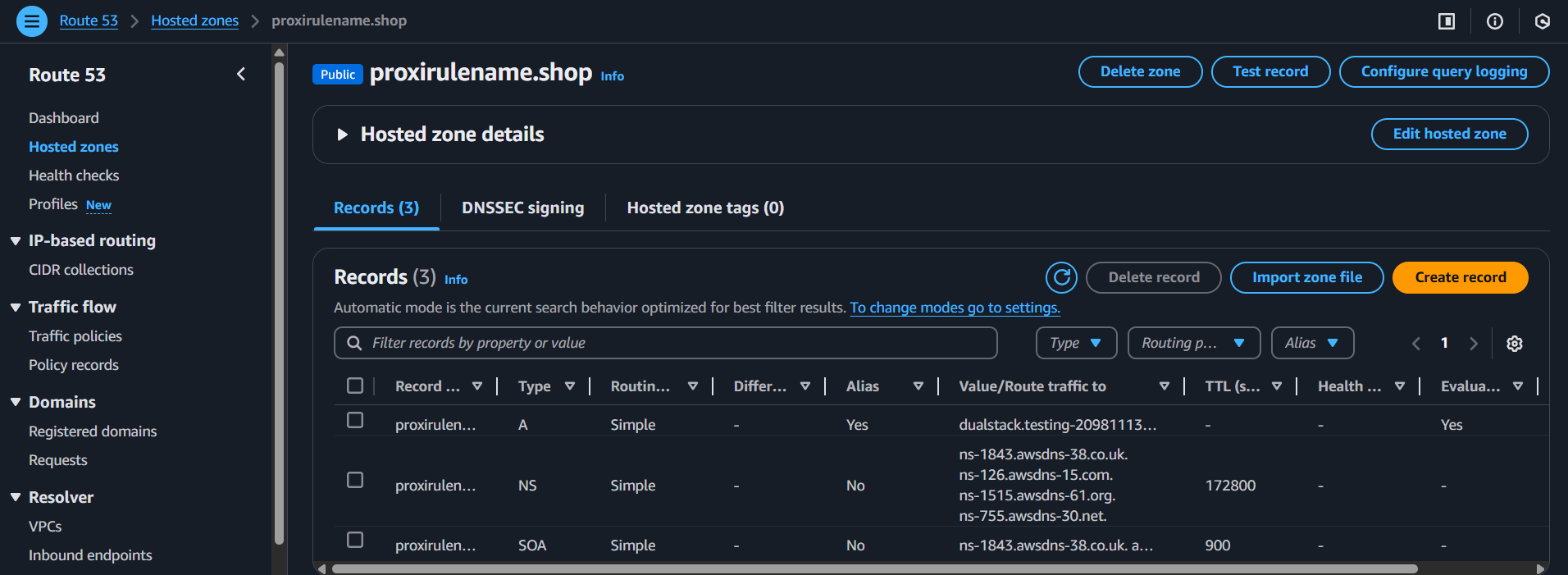
dashboard page as mentioned below:



**Map the docker machine to the Route 53 and use Load balancer to route the traffic.**

STEP:1

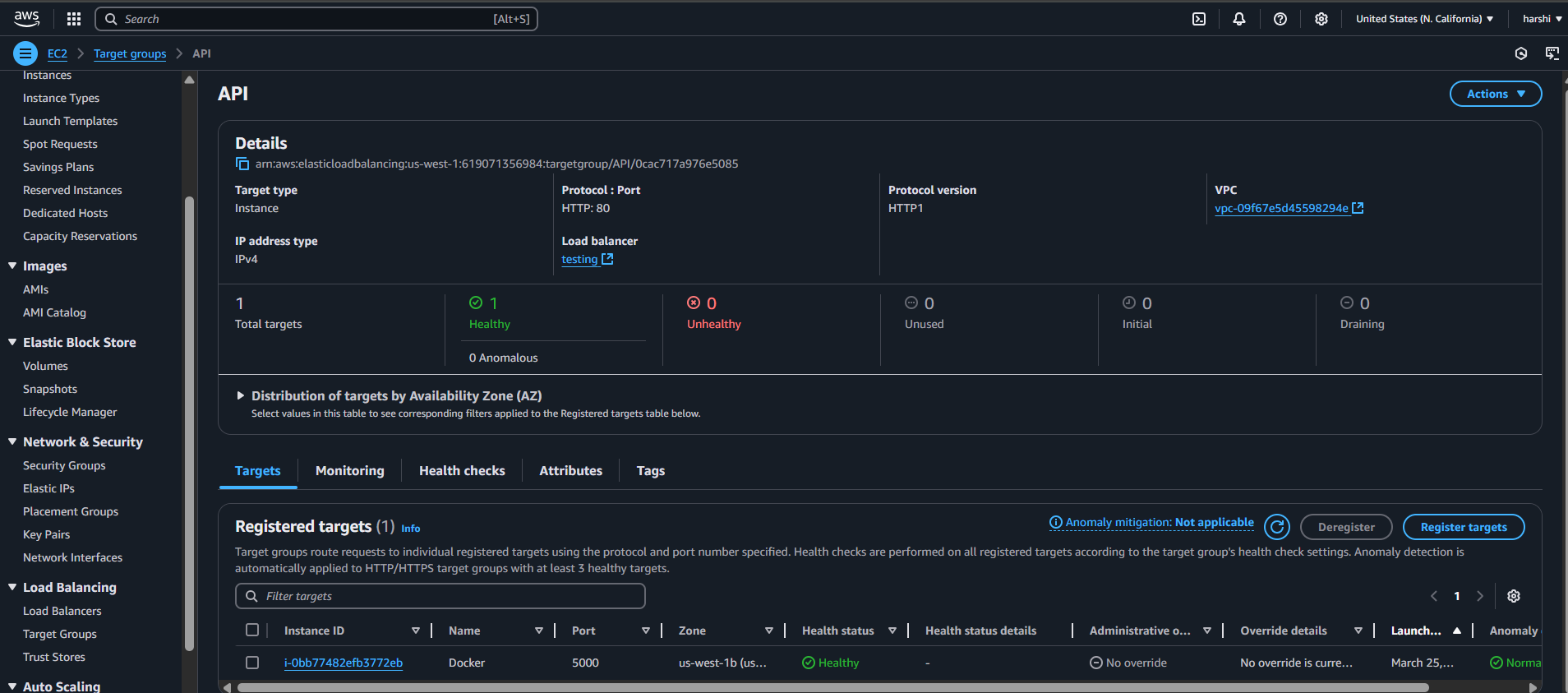
Create the route 53 and map the DNS as mentioned below:



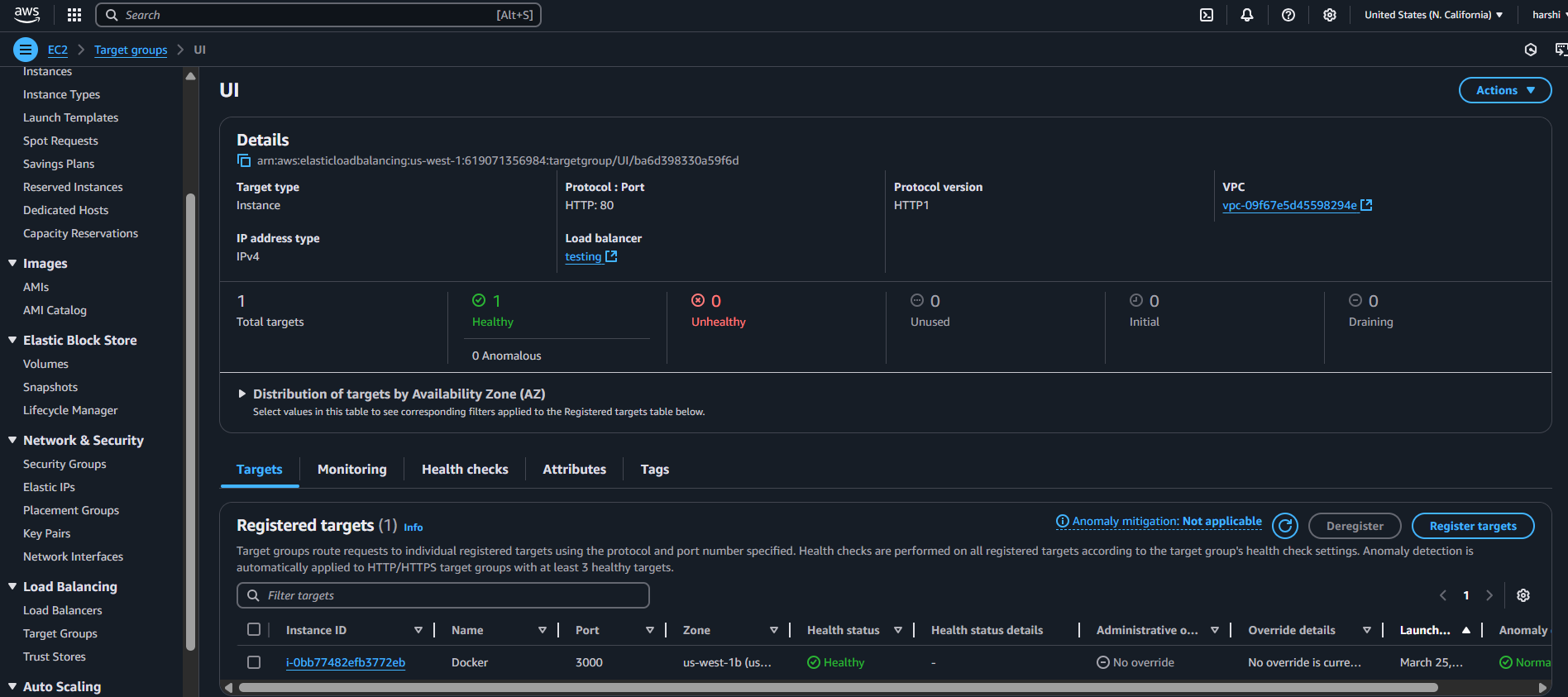
STEP 2:

Create target group with the docker machine for UI and API with 3000 and 5000 ports as mentioned below

api :

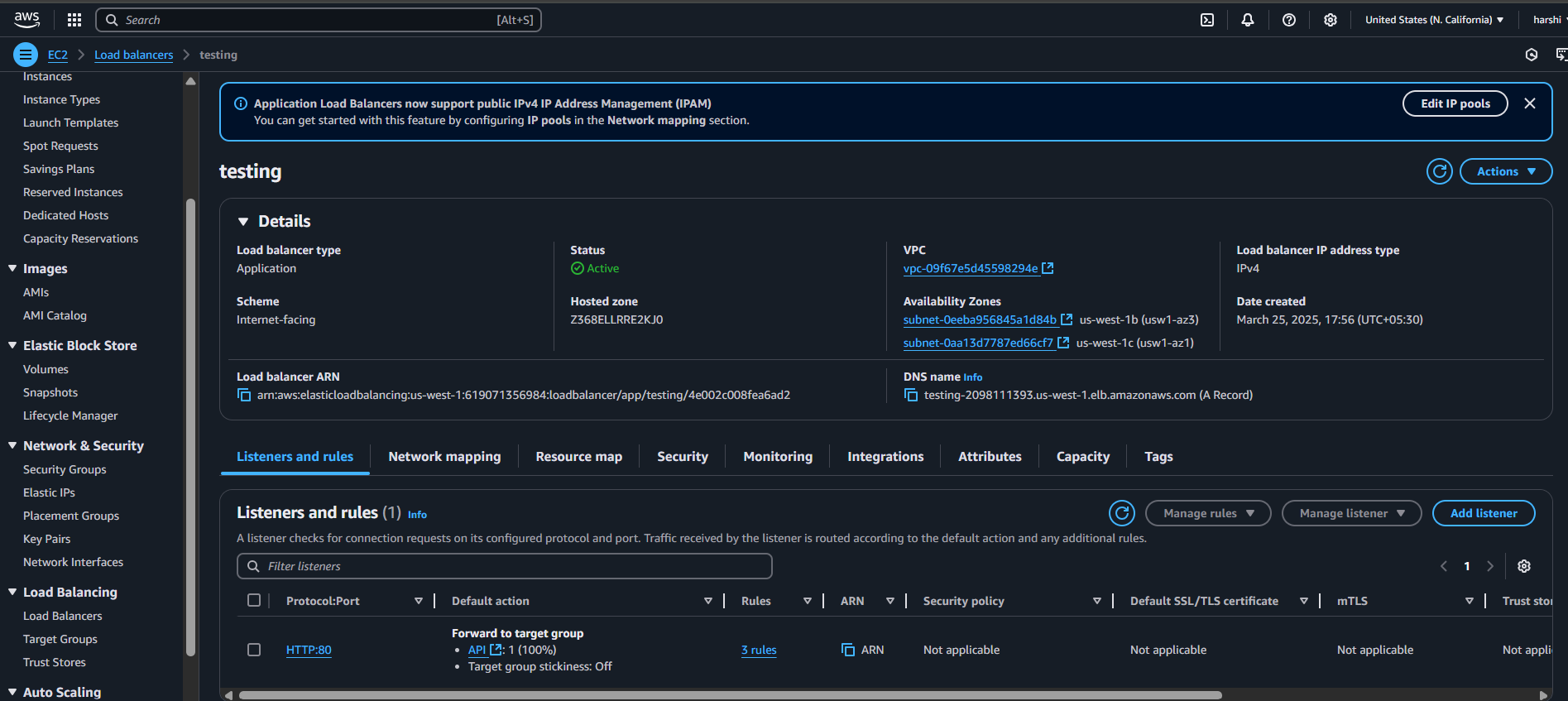


ui:

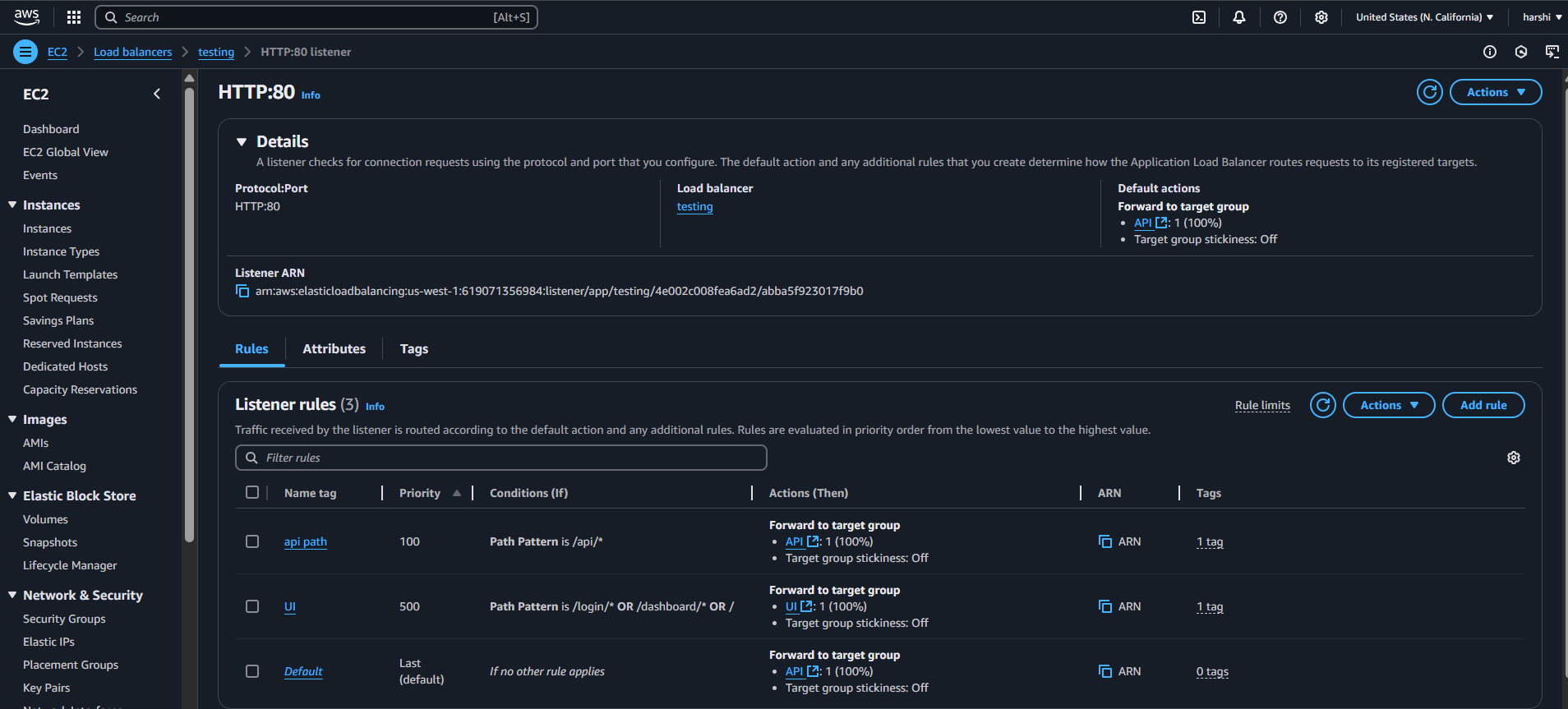


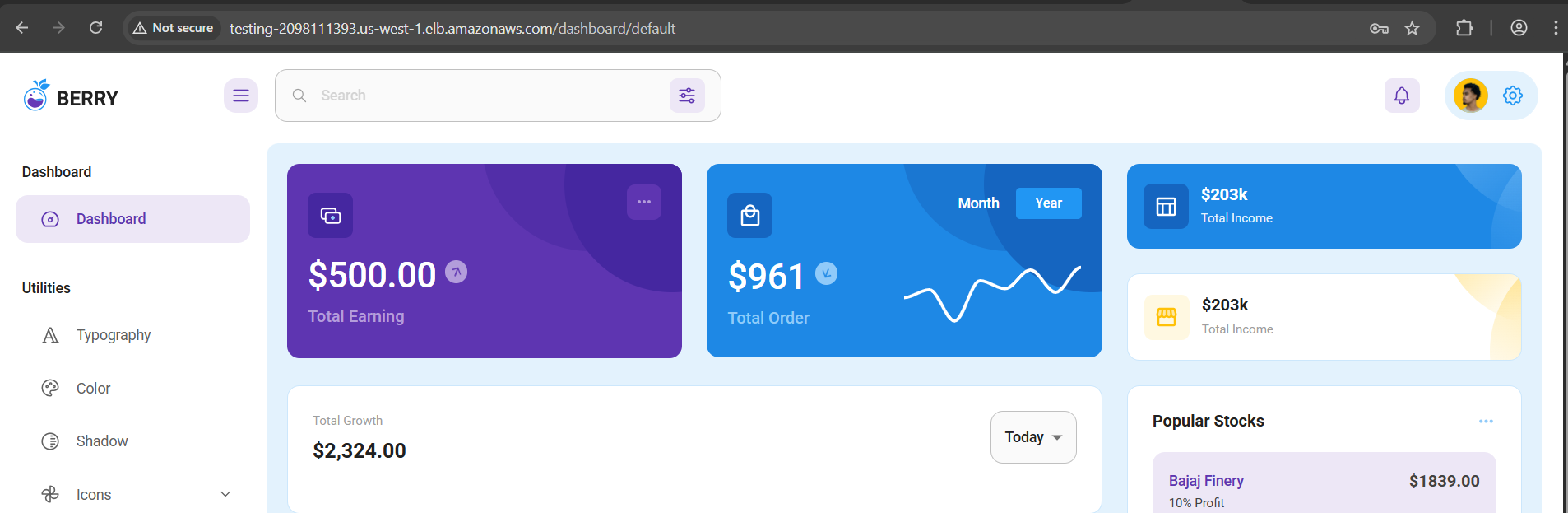
STEP 3:

Create load balancer and add these target groups as mentioned below:



Add the Listeners and rules as mentioned below:



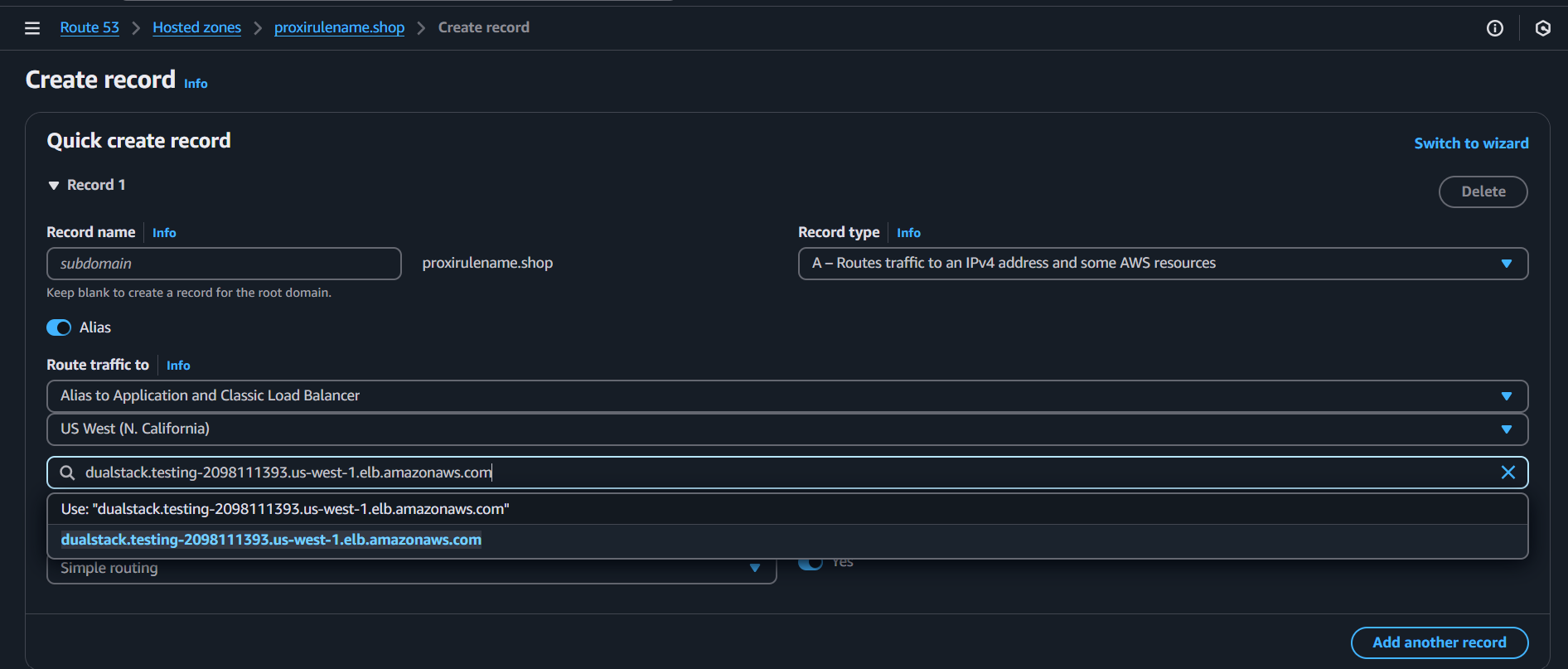


NOTE: Make sure that inbound rules are added to all traffic for the security groups which

you are adding to this docker machine.

STEP 4:

Add the loadbalancer to the Route 53 for the routing to happen:



STEP 5:

Use the dns name to check whether the route is happening or not:

