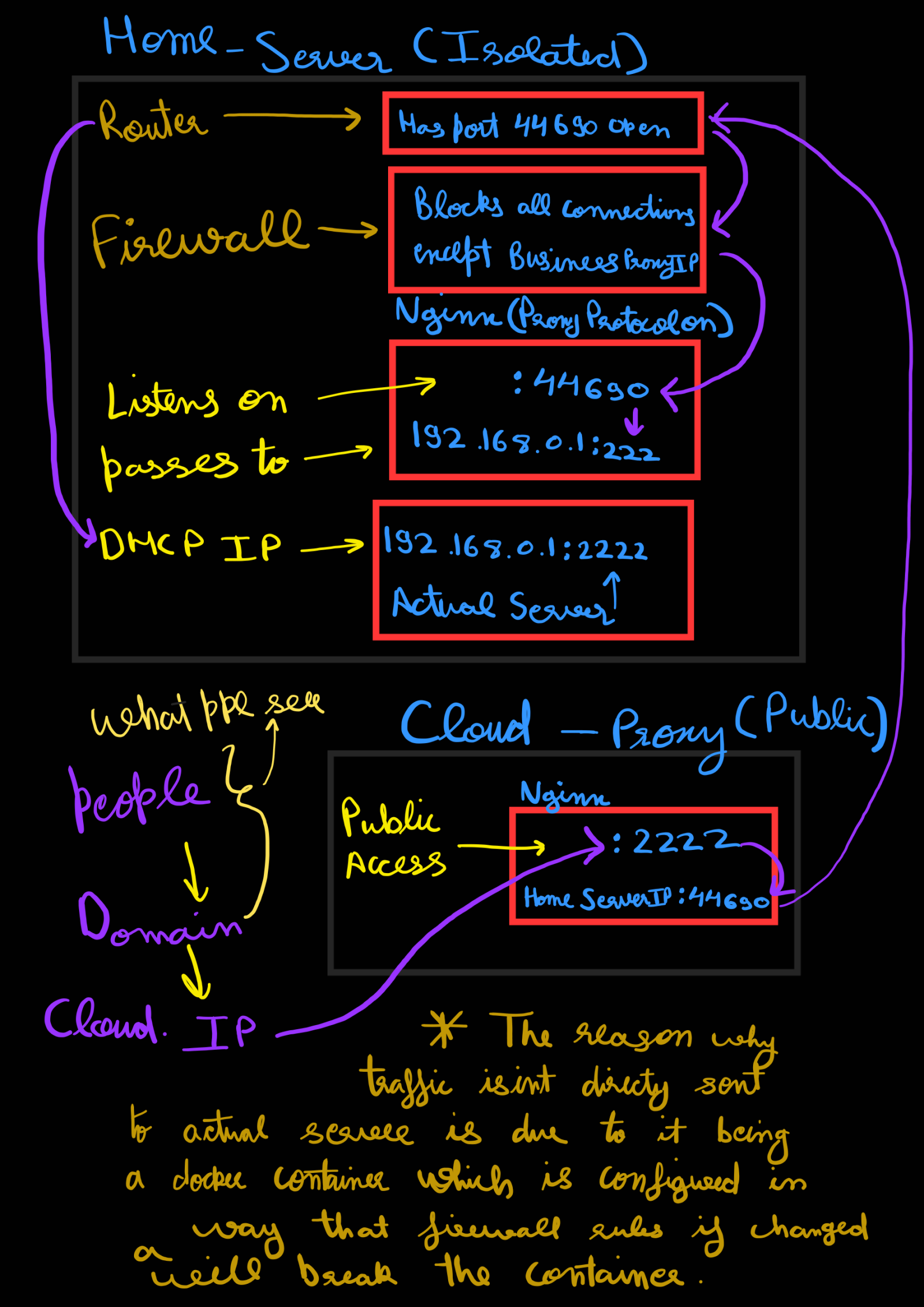
The Problem :-

Obfuscate a public facing self hosted servers that runs multiple different applications where inbound and outbound traffic is generated. (Around 20TB bandwith has been transfered till now.)

The Solution :-

Reverse Proxies!! I choose NGINX as it offered Load Balancing, Performance optmisation and flexibility in built

Next there is a diagram of the setup and my thought process. :)



* The reason why traffic isint directy sent to actual server is due to it being a docker container which is configured in a way that firewall rules if changed will break the container.

*If you noticed that the Business - Proxy has the same port as the Home Server, it is intentional and used for obfuscation

Now, the problem.

As it is a self host AKA it is hosted at someones home due to security reasons. The IP changed quite frequently. (Static IP was not possible)

The Solution,

ANSIBLE!!, Now if you are asking what is ansible, then in a nutshell it is a OPEN-Source automation tool that automates Various manual IT processes.

The Approach,

→ Set up ansible environment
→ Setting up SSH Keys
→ The Playbook!!

↓

Runs On Self Host Machine

- Step 1: Gathers facts
- Step 2: Gets the current IP Address of the machine
- Step 3: Saves it as a fact

Runs On Cloud Proxy

- Step 4: Gathers facts
- Step 5: Ensures all packages exist
- Step 6: Gets the IP of self host machine and saves it
- Step 7-10: Updates all configuration files with new IP's
- Step 11: Reloads NGINX

*If you took note of step 5. The playbook is designed with the though process that if the proxy machine is changed, no extra setup is required except setting up SSH keys!

A pretty neat detail hun :)