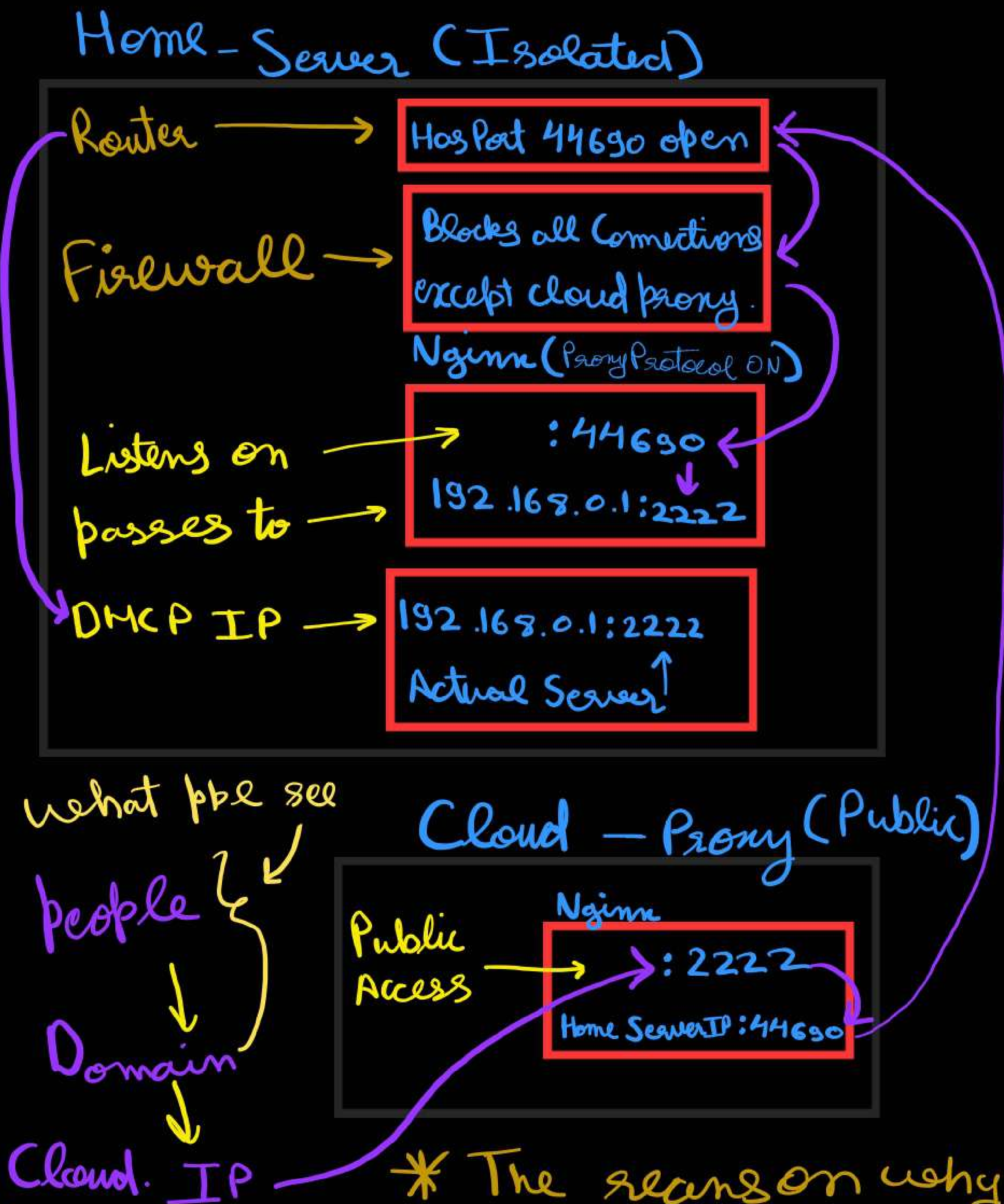
The Problem :-

Obfuscate a public facing business server that is located on premise that runs multiple different applications where inbound and outbound traffic is generated. (Around 20 TB bandwith has been transfered till now.

The Solution :-

Reverse Proxies!! I choose NGINX as it offered Load Balancing, performance optimisation and flexibility built in.

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



* The reason on why traffic isn't directly sent to the actual application is due to it being in a docker container which is configured in a way that if firewall rules are applied on the container it will break.

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

<u>Now, the problem.</u>

As it is a on premise machine AKA hosted on a business location due to security reasons. The IP changed quite frequently. (Static IP was not possible)

<u>The Solution,</u>

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

The Approach,

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

Runs On Premise
- 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 On premise-machine and saves it
- Step 7-10: Updates all config files with the new IP's
- Step 11: Reloads NGINX

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

A pretty neat detail hun :)

If you have read this far, thank you from the bottom of my heart for sticking with me. I truly appricate you being here.