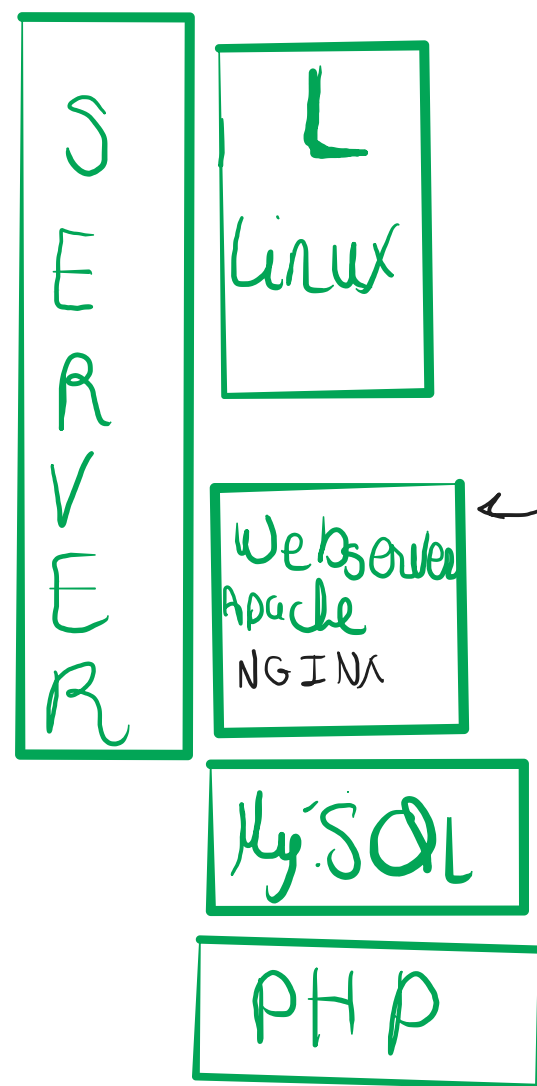


# LAMP STACK Architecture

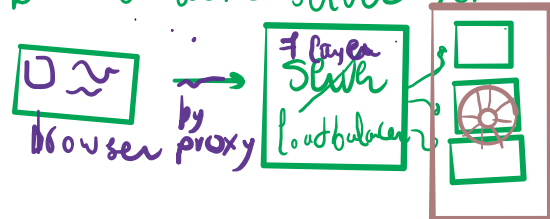


middle man between browser & website content  
static info  
VS Application Servers  
handle dynamic content & compile  
in a language like java or python

## Apache Vs NGINX

HTTP & reverse proxy servers

browser asks server for a webpage



- 1) Load balancing
- 2) Security
- 3) Caching
- 4) Compression
- 5) SSL

Apache came first <sup>extensible</sup> <sup>HTTP traffic</sup>  
web reverse proxy-server <sup>edge in dynamic</sup>  
NGINX <sup>out performs Apache</sup>  
Last simple Config for static

How does it work?

- webserver sends request (front page of a website)
- then hits Apache (webserver) → run PHP → send SQL DB → Linux → packets → web browser

## Downsides:

- SPOF If the single server fails the website goes down
- Downtime for maintenance for deploying new code on NGINX website goes down for a while
- Scalability not good with high traffic volumes