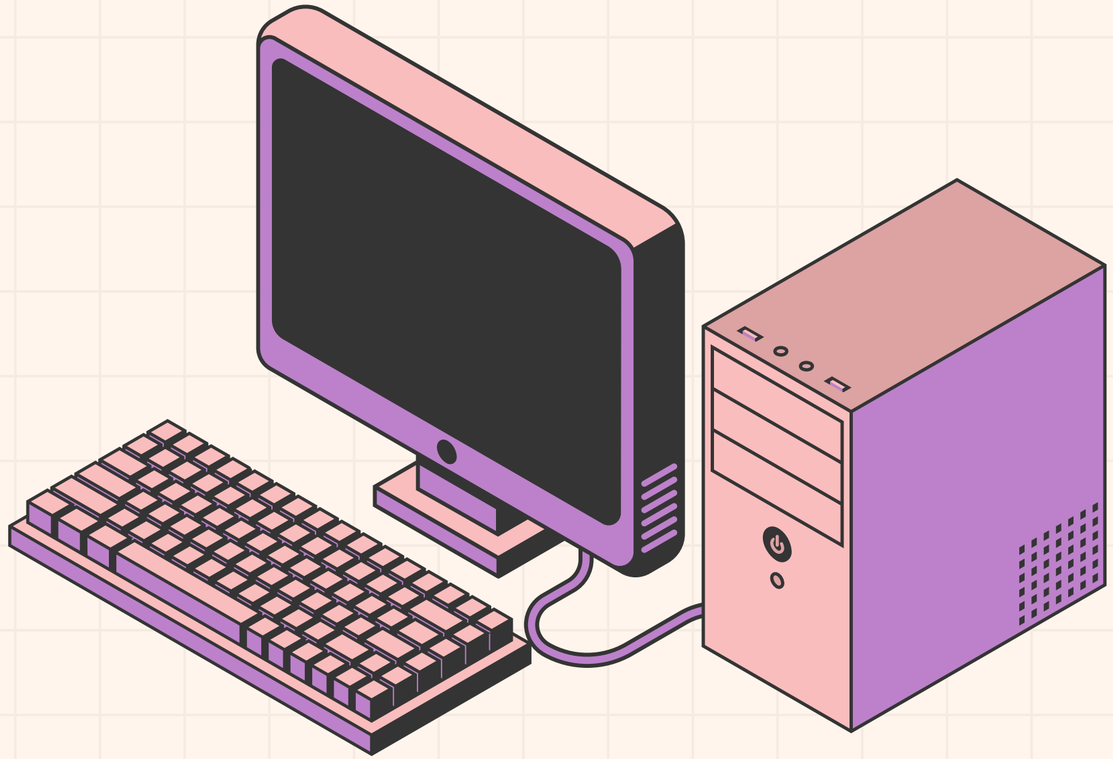
NHÓM 4

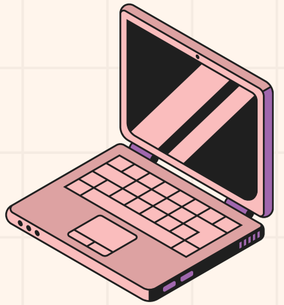
CONTENTS


NGINX®



WHAT IS NGINX ?

- Free
- Open source
- Developed by a Russian developer – Igor Sysoev
- Public release in 2004



WHO USE NGINX ?

AUTODESK

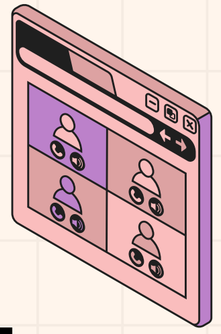
ATLASSIAN

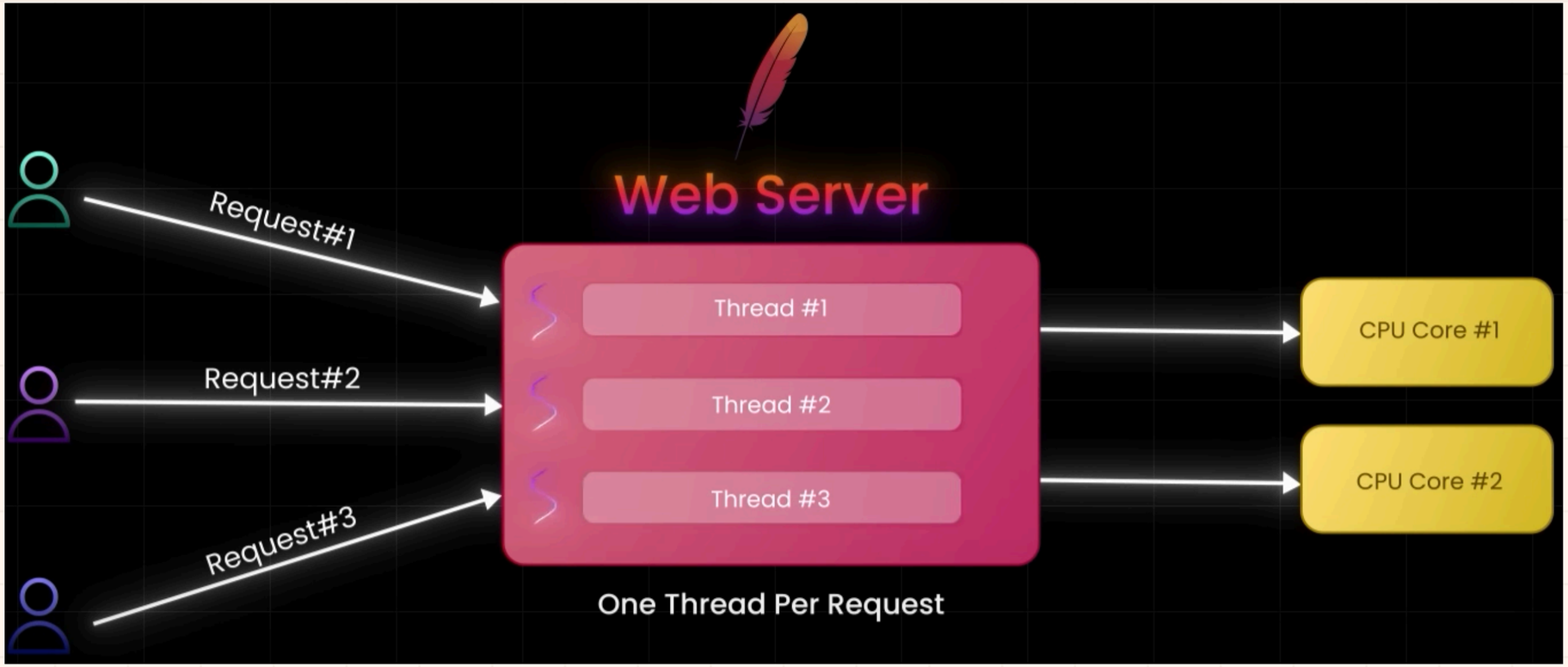
GitLab

Microsoft

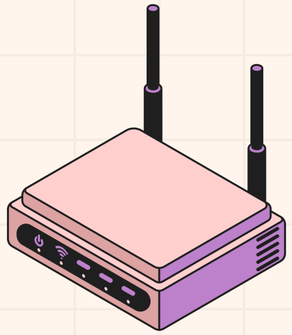
Google

ARCHITECTURE AND PRINCIPLE OF WORKING





ARCHITECTURE AND PRINCIPLE OF WORKING



EVENT-DRIVEN

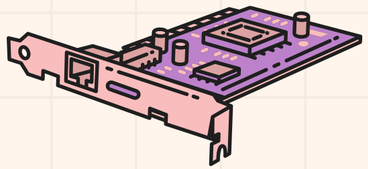
It will respond to actions generated by users or the system.

ASYNCHRONOUS

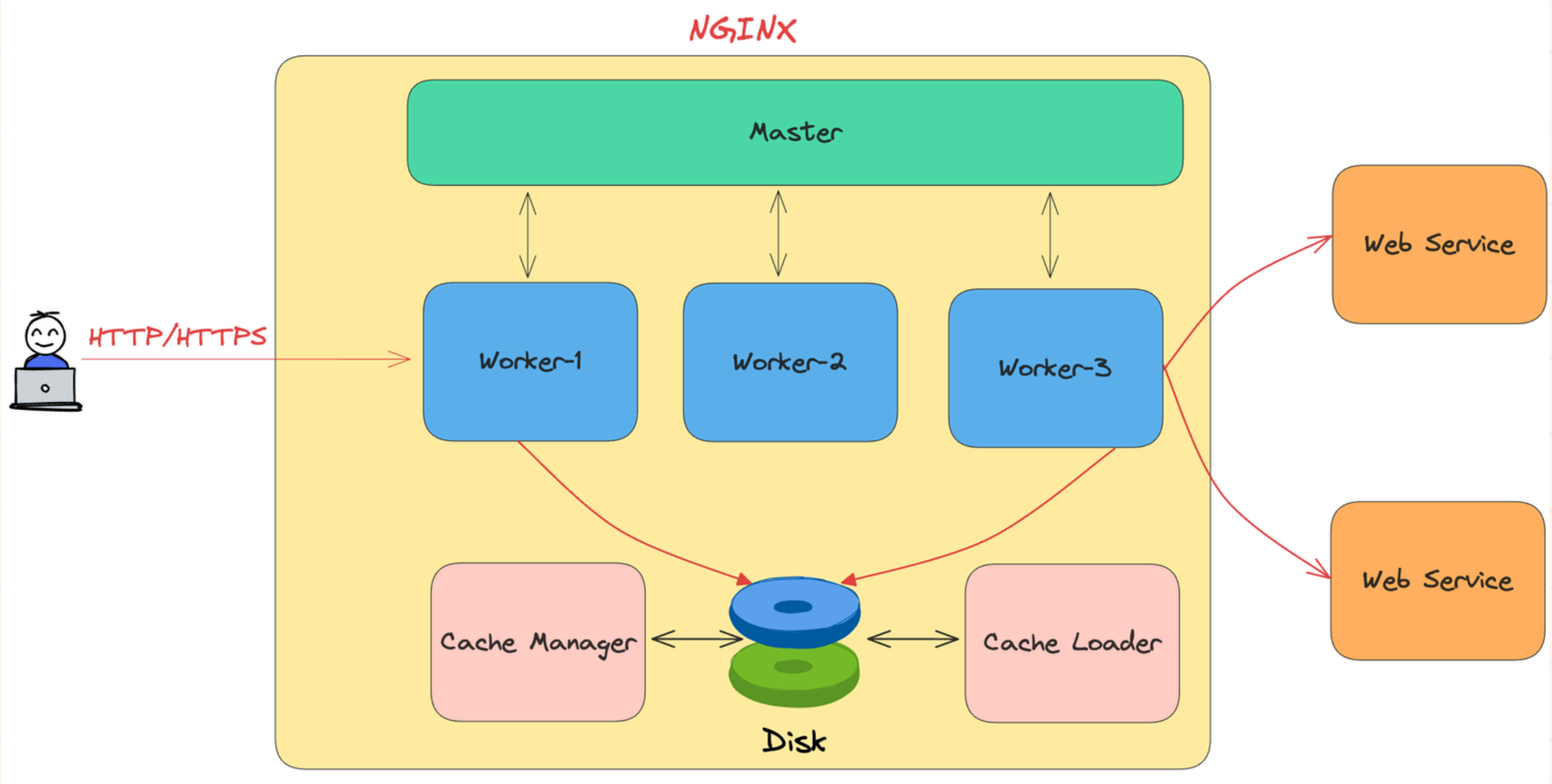
The system will not depend on the arrival time of signals to operate.

NON-BLOCKING

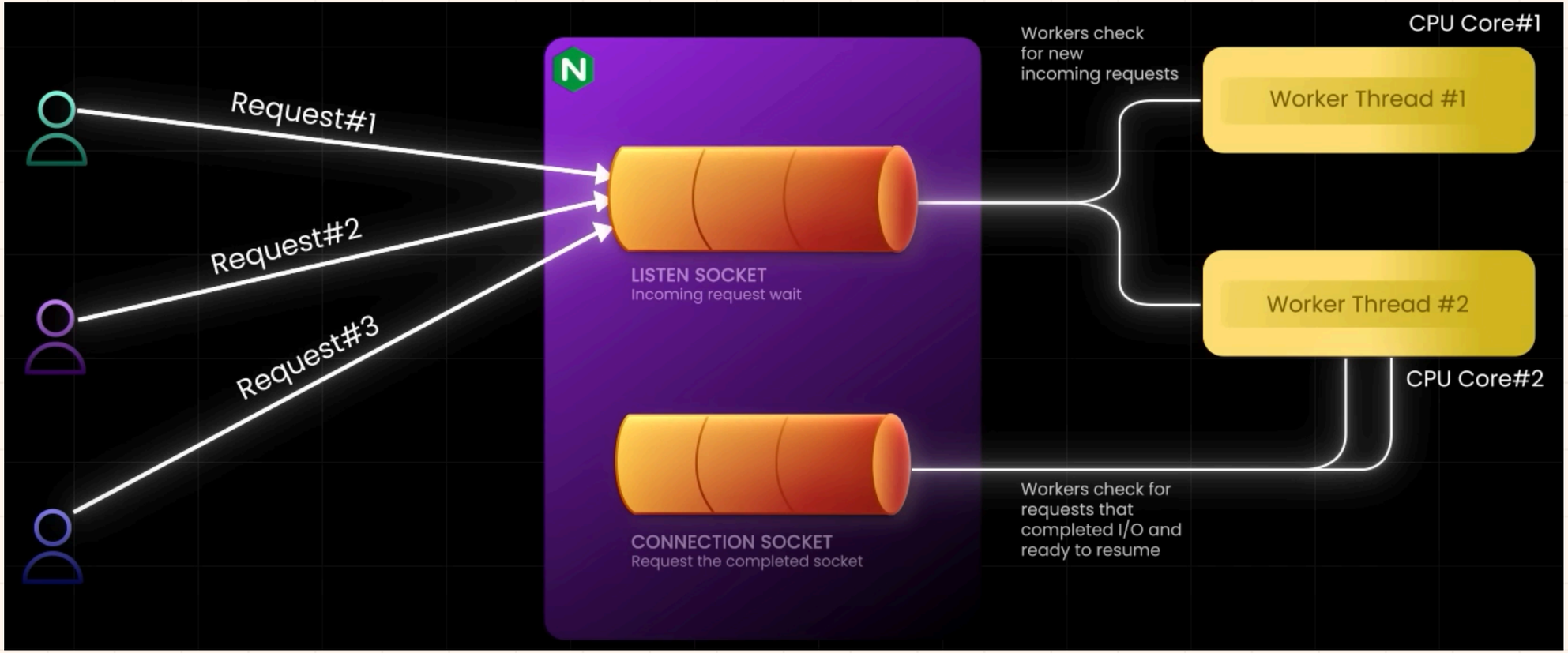
Functions will not stop waiting for something to happen.



ARCHITECTURE AND PRINCIPLE OF WORKING

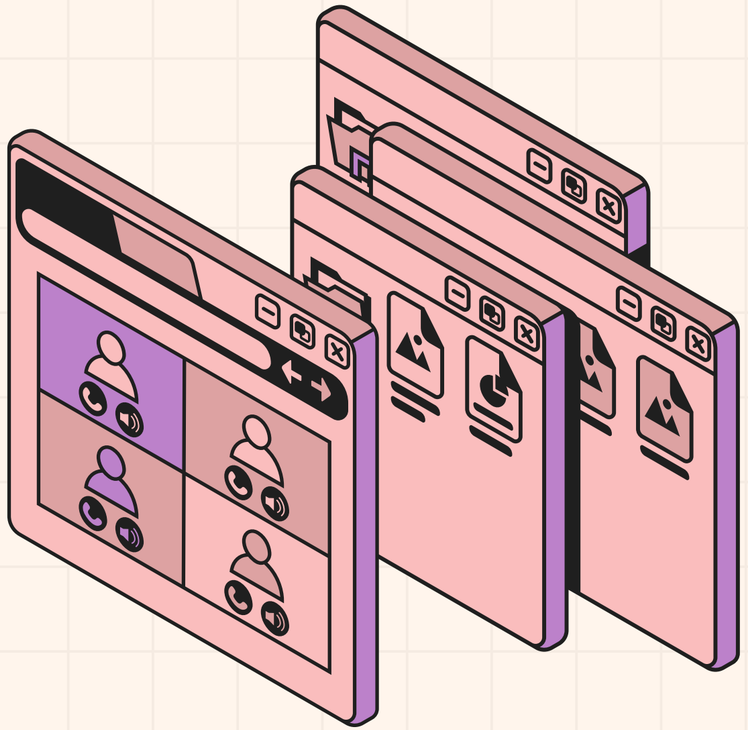


ARCHITECTURE AND PRINCIPLE OF WORKING



IMPORTANT CONFIGURATION FILES

- **/etc/nginx/:** Main configuration directory for the Nginx server, acting as the root location where all configuration files are stored.
- **/etc/nginx/nginx.conf:** This is the main configuration file for Nginx. This file contains global settings that affect the entire server.
- **/var/log/nginx/:** This directory is the default location for Nginx log files.

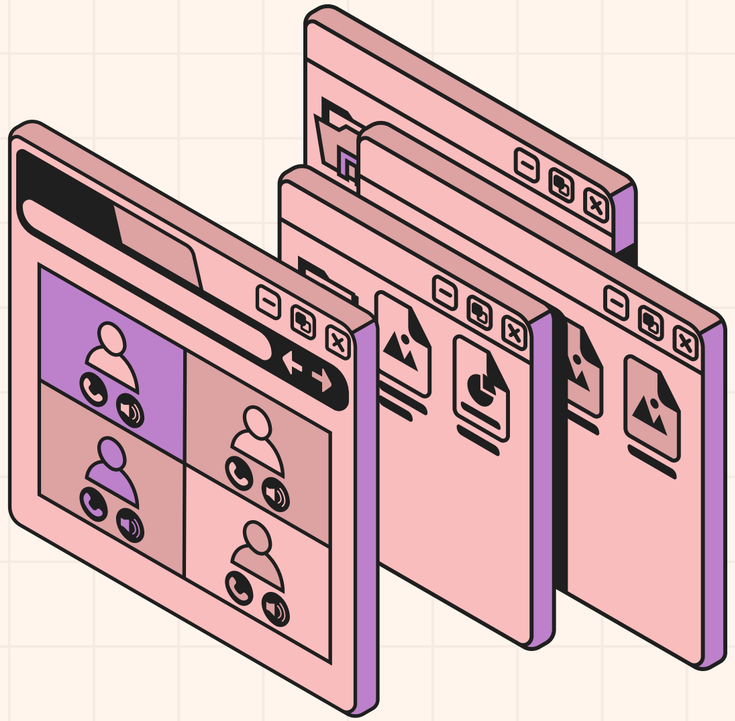


IMPORTANT CONFIGURATION FILES

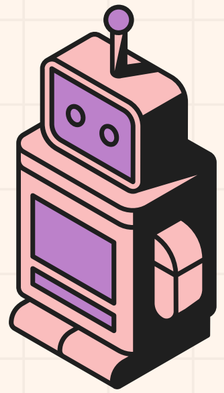
- **/etc/nginx/sites-available:** used to store available configuration. Contains all configuration files of websites/server blocks that you can run with Nginx.

- **/etc/nginx/sites-enabled:** used to store the currently enabled configuration. Contains symbolic links pointing to the actual configuration files in sites-available.

- **/var/www/html:** is the default directory containing the application's source code, it can also be placed in any other directory.



/ETC/NGINX/NGINX.CONF
EVENT CONTEXT

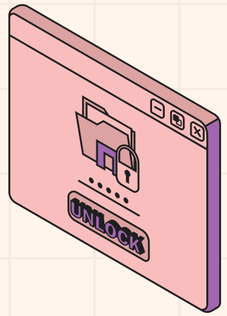


- Used to set global options that affect how Nginx handles connections at the global level.
- There can only be one event context defined in the Nginx configuration file.
- Nginx uses an event-driven connection handling model ==> the directives defined here will determine how worker processes handle connections.
- The directives found here are used to select which connection handling technique to use or to modify how these methods are implemented.

```
events {
        worker_connections 768;
        # multi_accept on;
}
```

/ETC/NGINX/NGINX.CONF

HTTP CONTEXT

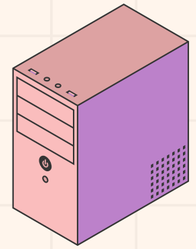


- This context will contain all the directives and other contexts needed to determine how the program will handle an HTTP or HTTPS connection.

- The http context is a sibling of the event context, so they should be listed side by side, rather than nested. Both are children of the main context

```
http {

        ##
        # Basic Settings
        ##

        sendfile on;
        tcp_nopush on;
        types_hash_max_size 2048;
        # server_tokens off;
```

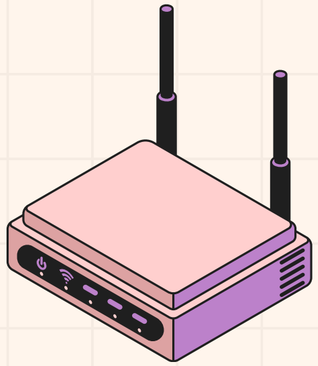
/ETC/NGINX/SITES-AVAILABLE
SERVER CONTEXT



- In 1 HTTP context, there can be multiple server contexts, because each server context is an instance that defines a specific virtual server to handle client requests.

- Server context is a child of HTTP context.

```
server {
        listen 82;
        server_name myapp1.com;
        root /var/www/myapp1;
        index index.html index.htm;
        location / {
                try_files $uri $uri/ =404;
        }
}
```



/ETC/NGINX/SITES-AVAILABLE
LOCATION CONTEXT

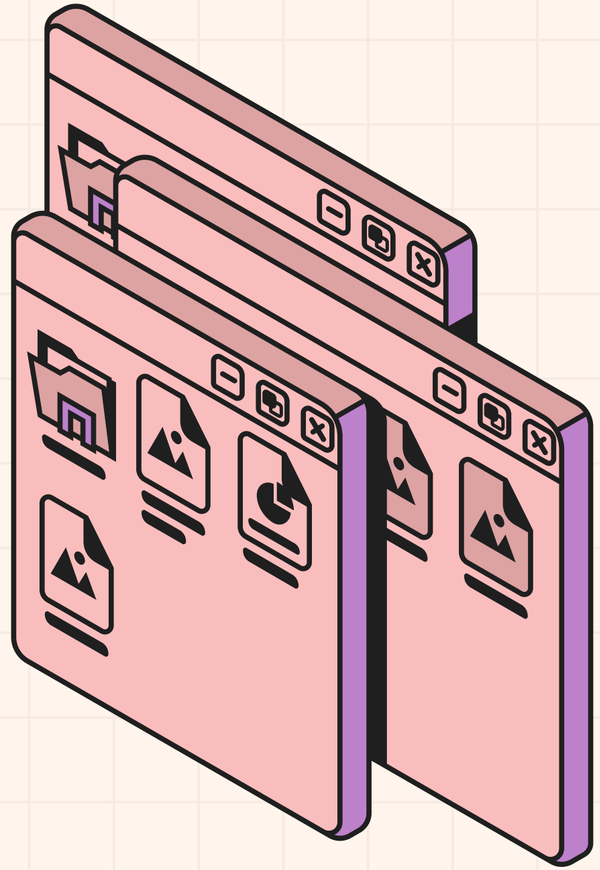
- Each location is used to handle a certain type of request from the client

- Location contexts reside in server contexts and are used to decide how to handle URI – Uniform Resource Identifiers

```
location / {

    proxy_pass http://192.168.1.4:5000;  # Gunicorn

    proxy_set_header Host $host;

    proxy_set_header X-Real-IP $remote_addr;

    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;

    proxy_set_header X-Forwarded-Proto $scheme;

}
```

OTHERS IMPORTANCE

- **worker_processes:** defines the number of worker processes.
- **worker_connections:** is the maximum number of connections that each worker process can handle simultaneously.
- **access.log & error.log:** Used to log errors that occur in Nginx and also used for debugging purposes.
- **gzip:** This is the setting for gzip compression on Nginx responses.

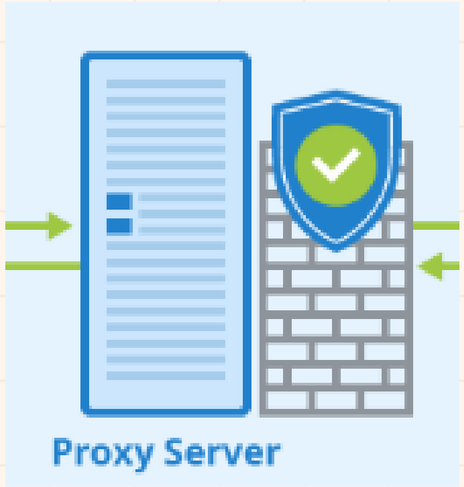


SOME COMMON COMMANDS

Command	Description
nginx -h	Shows the NGINX help menu.
nginx -v	Shows the NGINX version.
nginx -V	Shows the NGINX version, build information, and configuration arguments
nginx -t	Tests the NGINX configuration.
nginx -T	Tests the NGINX configuration and prints the validated configuration to the screen
nginx -s signal	The -s flag sends a signal to the NGINX master process. You can send signals such as stop , quit, reload and reopen
systemctl start nginx	Start the nginx process , can also use systemctl restart nginx
systemctl stop nginx	Stop nginx Process
systemctl status nginx	Show the nginx status either running or failed

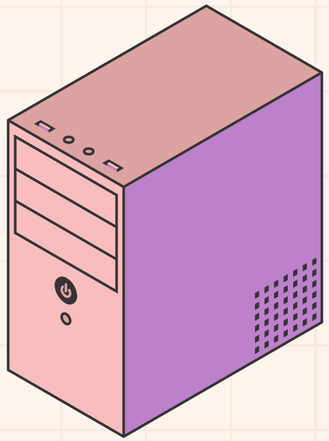
WHAT IS NGINX USED FOR ?

WEB SERVER

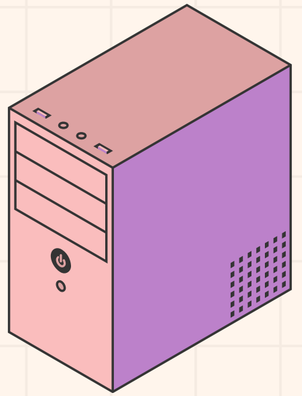


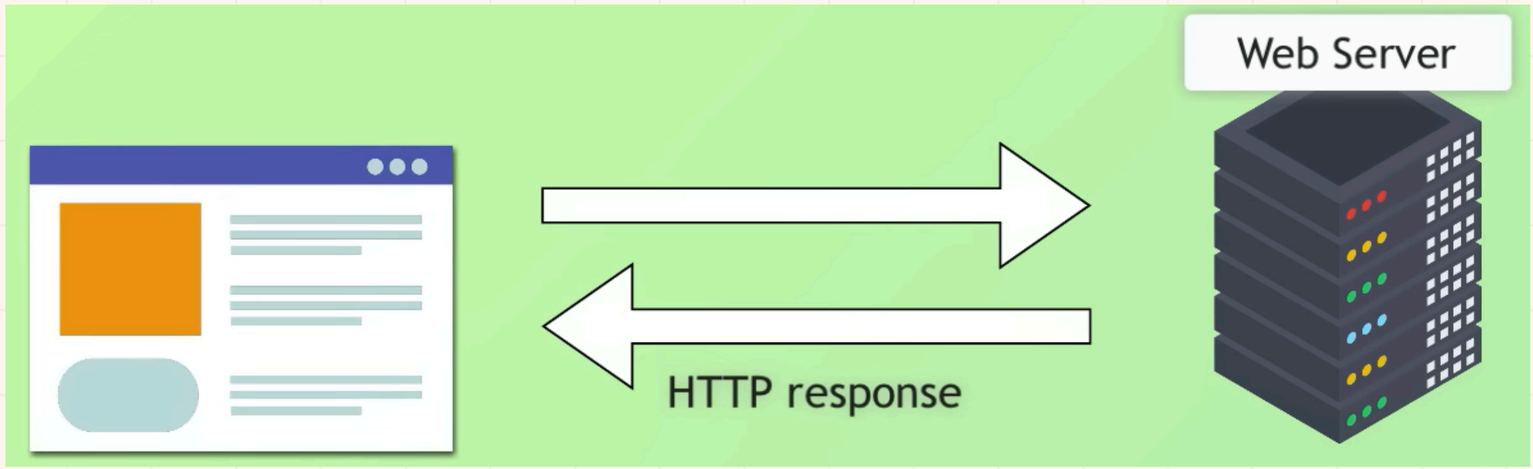
PROXY SERVER

WEB SERVER

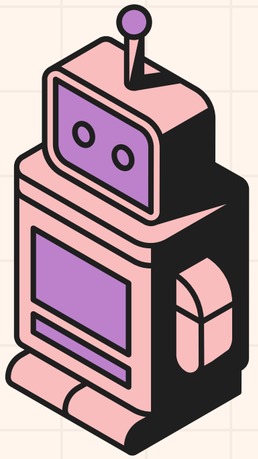


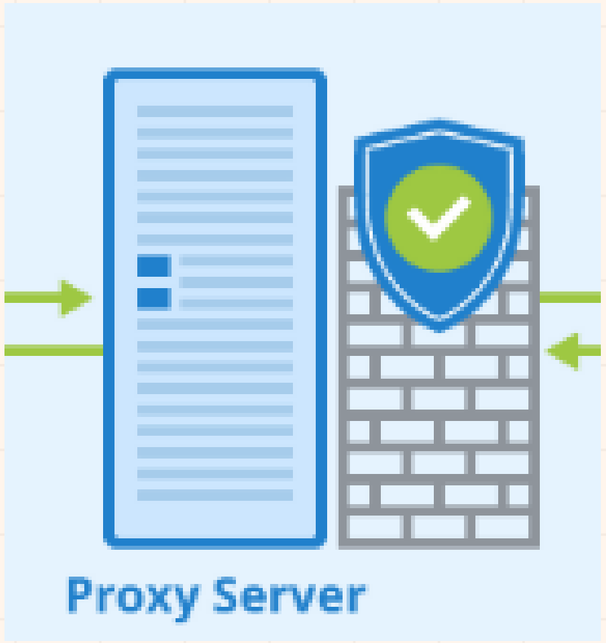
WEB SERVER





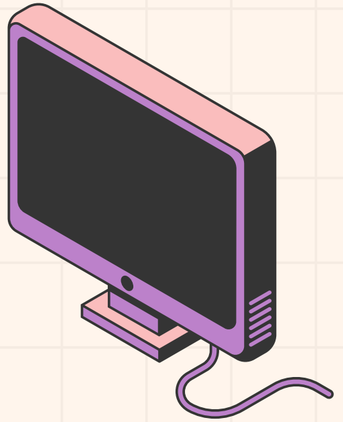
PROXY SERVER

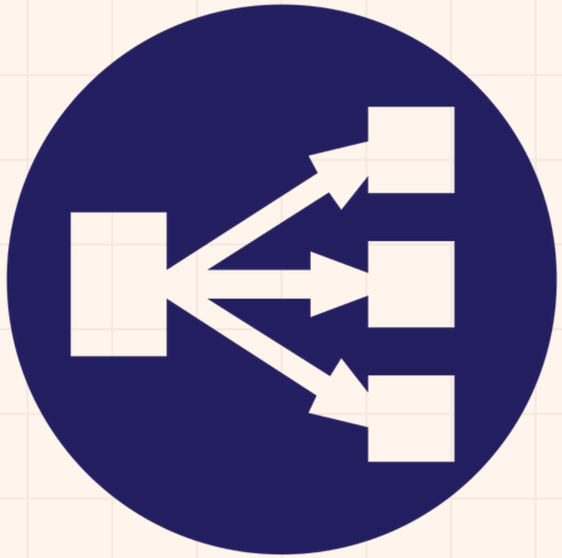




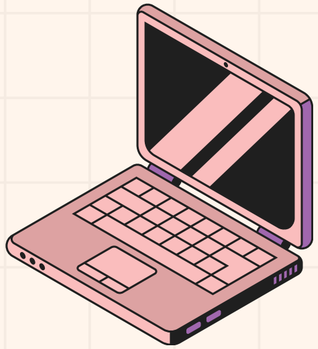
PROXY SERVER

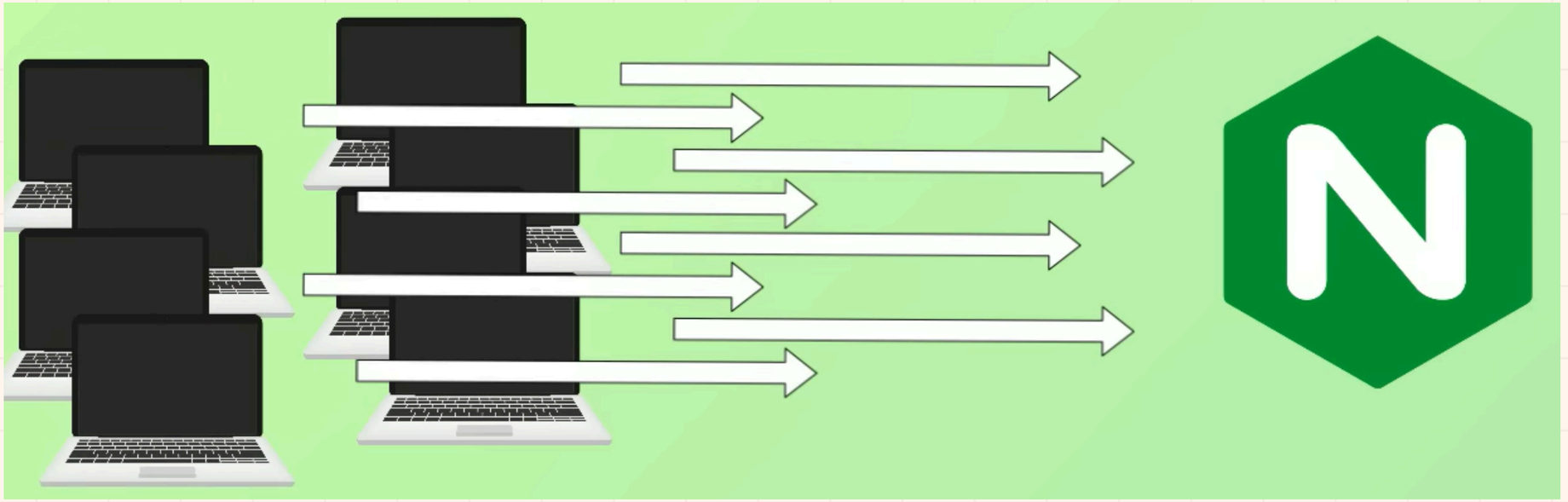
LOAD BALANCER





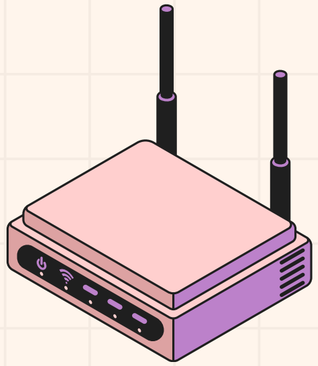
PROXY SERVER
LOAD BALANCER

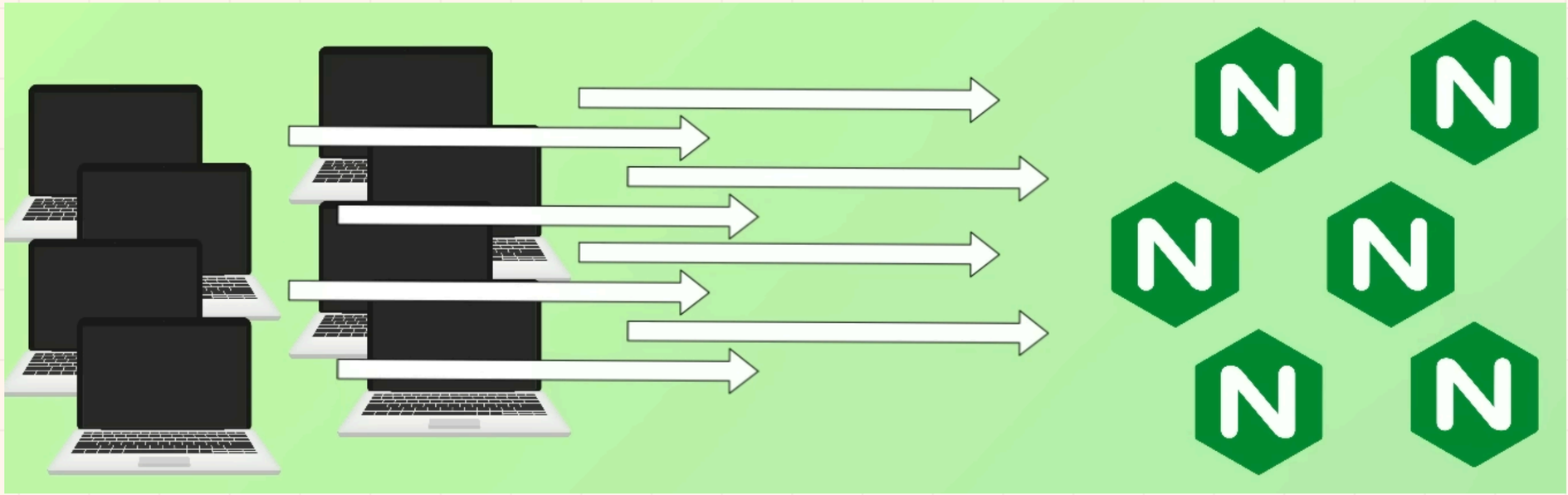




PROXY SERVER

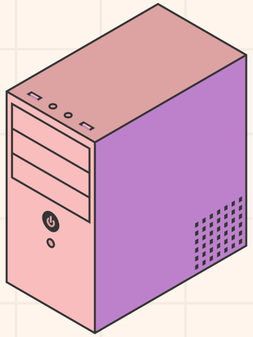
LOAD BALANCER

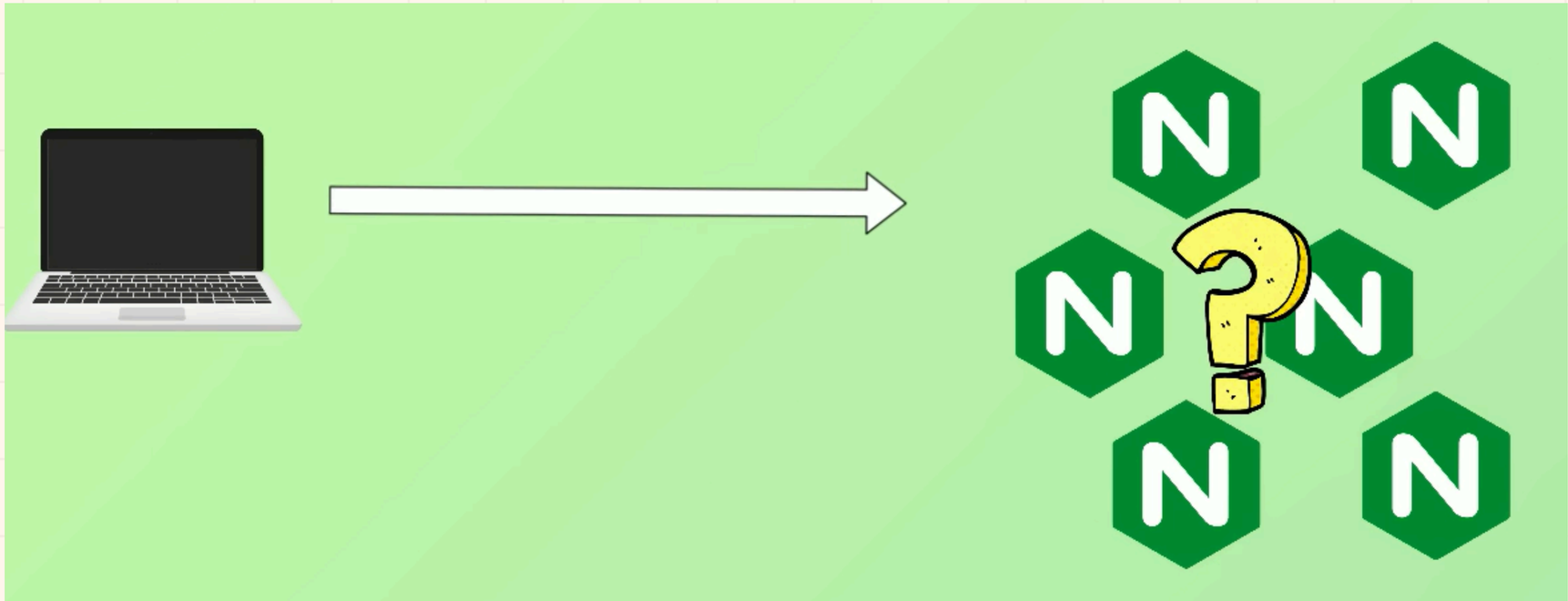




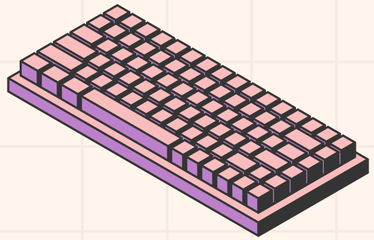
PROXY SERVER

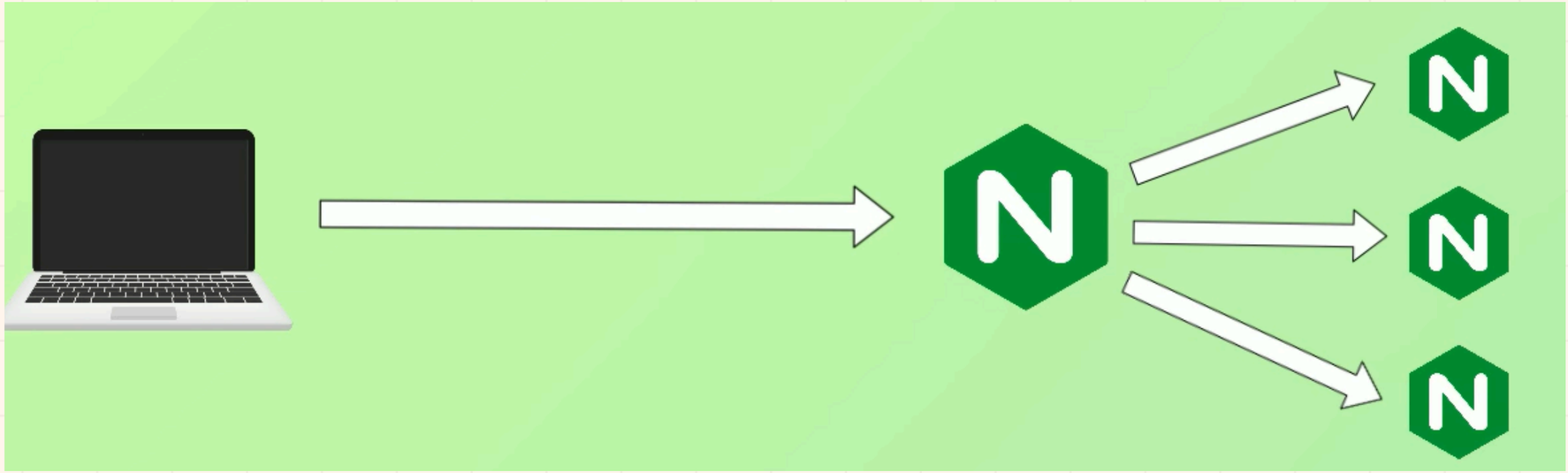
LOAD BALANCER





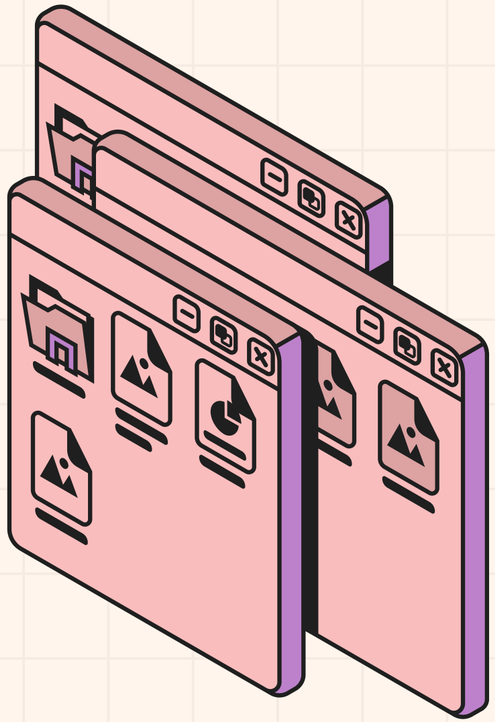
PROXY SERVER
LOAD BALANCER





REQUEST DISTRIBUTION ALGORITHM

- **Round-Robin:** this is the default algorithm of Nginx. Use the round-robin algorithm to decide which server to forward the request to.
- **Least-Connected:** this algorithm will forward the request to the server that currently has the least connections.
- **IP-Hash:** this algorithm will use the client's IP address to decide which server the client's request will be sent to.



BENEFITS OF LOAD BALANCER

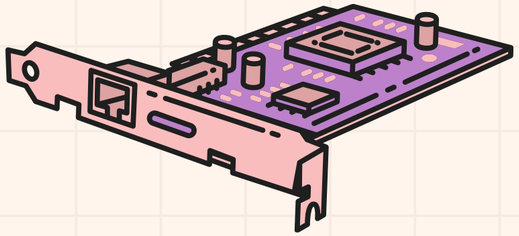
SYSTEM LOAD BALANCING

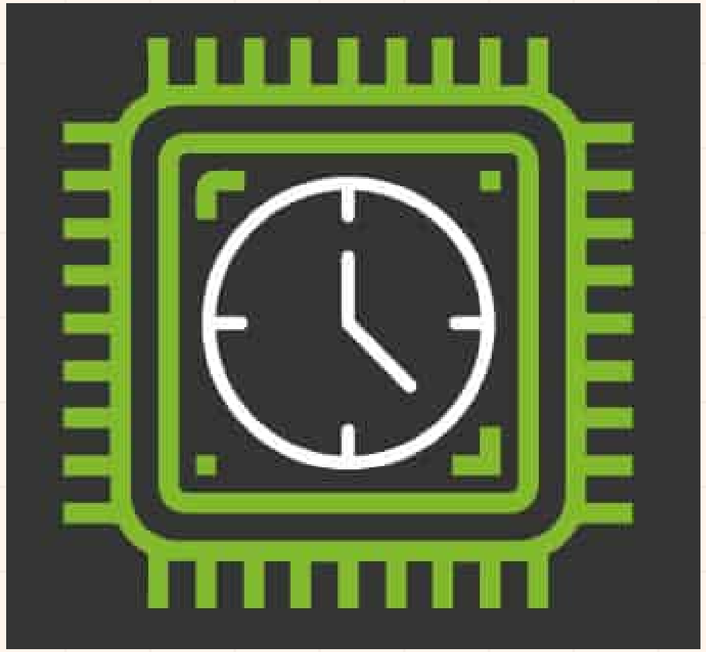
INCREASED FAULT TOLERANCE

EASY SCALING

PROXY SERVER

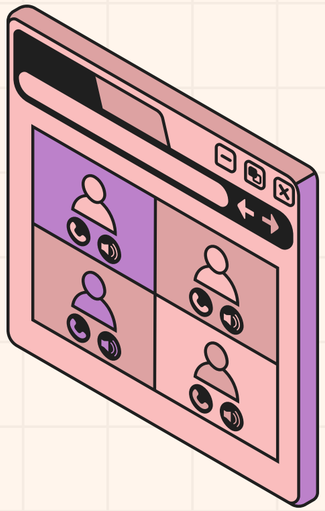
CACHE

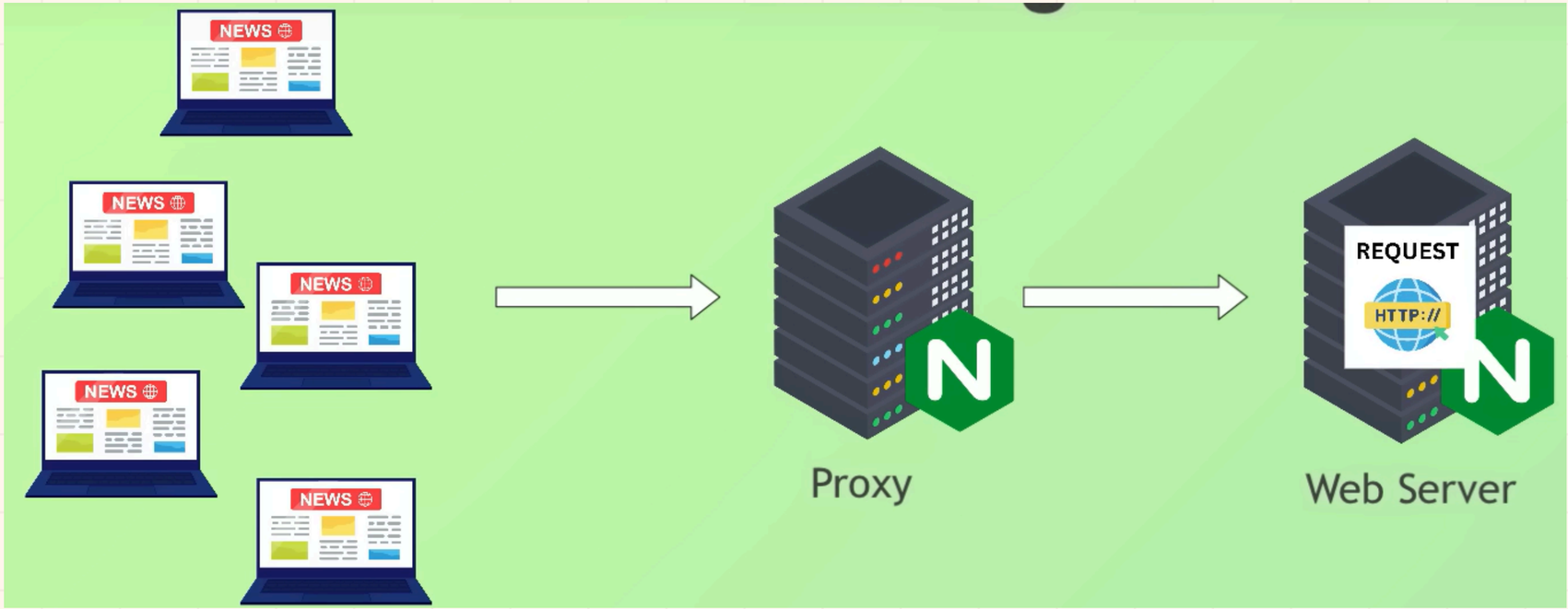




PROXY SERVER

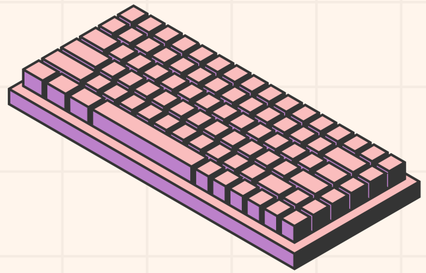
CACHE

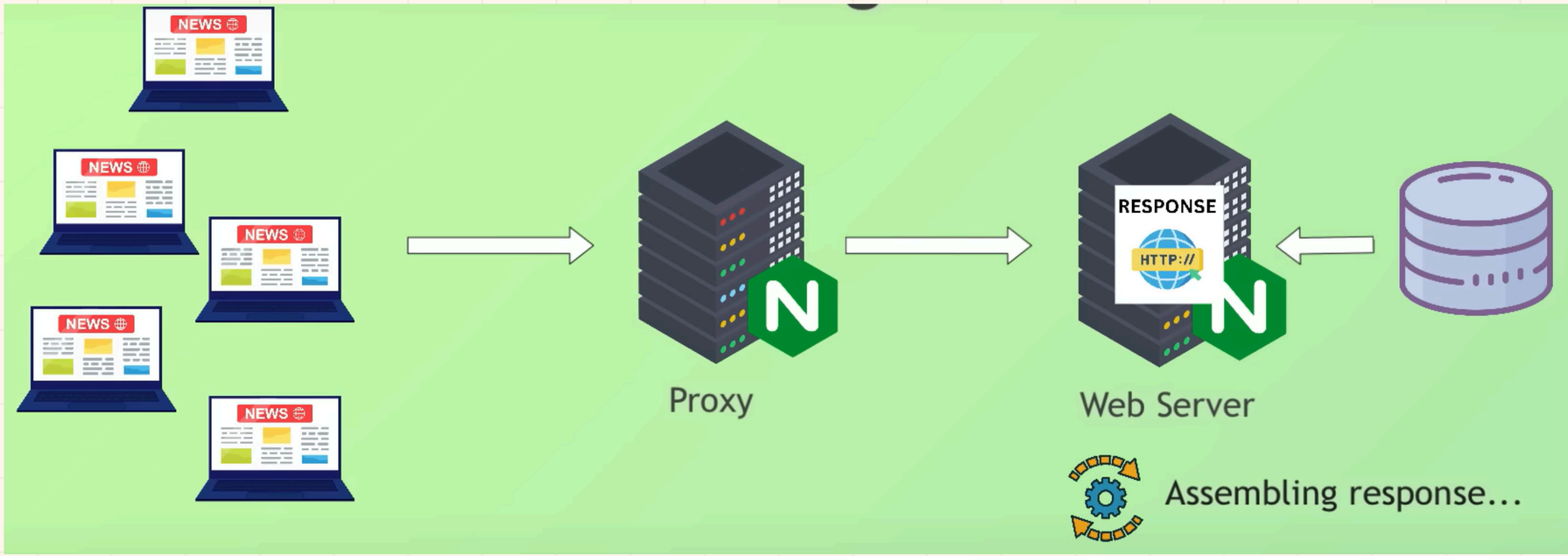




PROXY SERVER

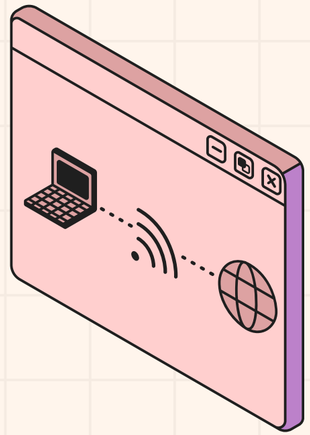
CACHE

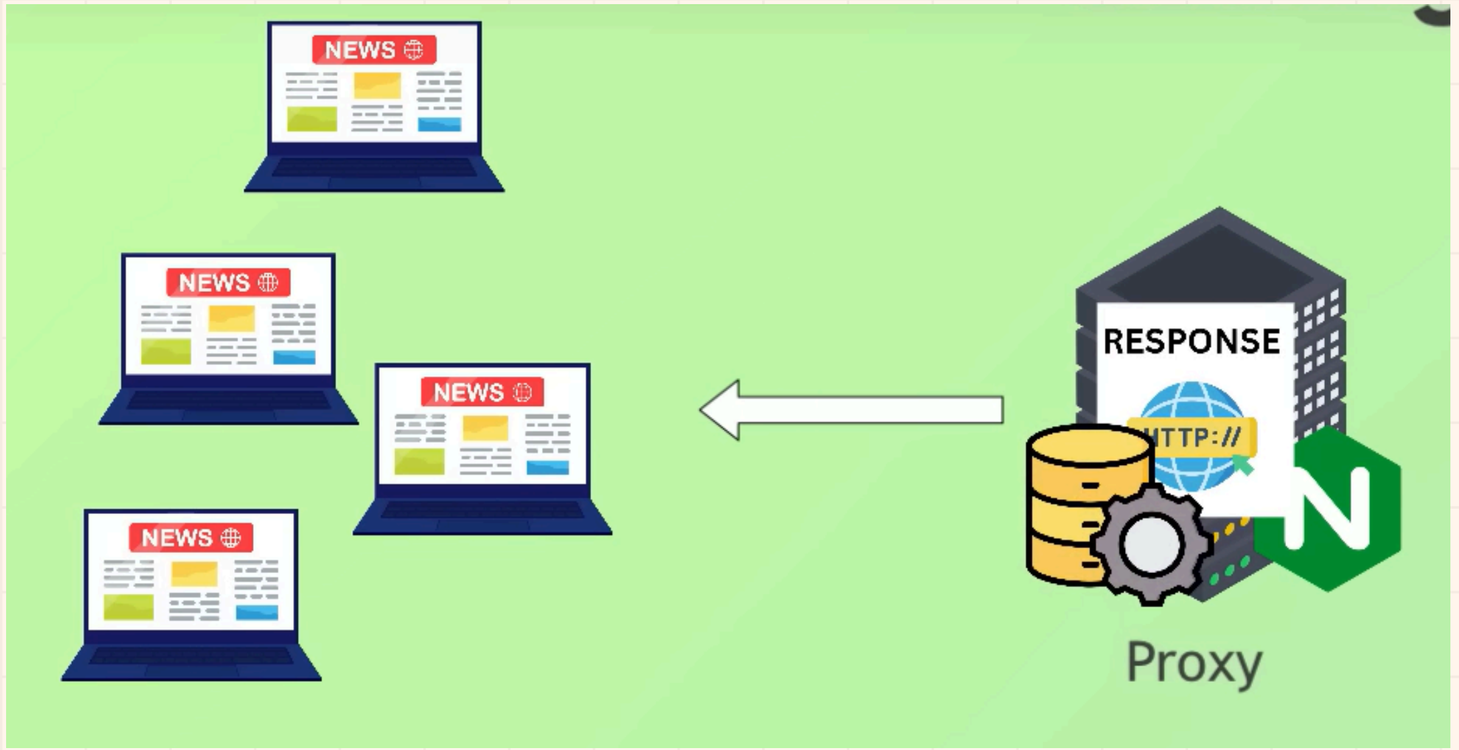




PROXY SERVER

CACHE





BENEFITS OF CACHE

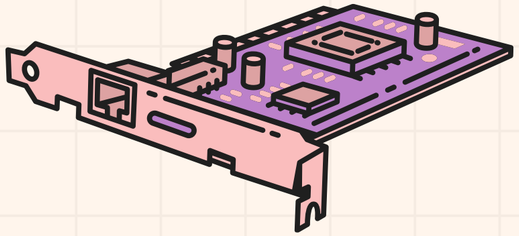
INCREASE
RESPONSE SPEED

REDUCE
BACKEND LOAD

OPTIMIZE
SEO & UX

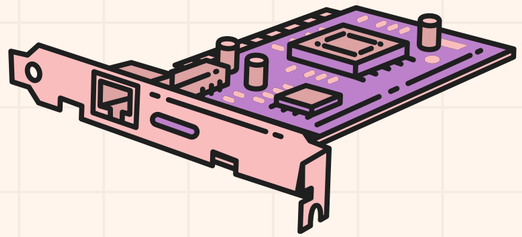
PROXY SERVER

ONE ENTRYPOINT



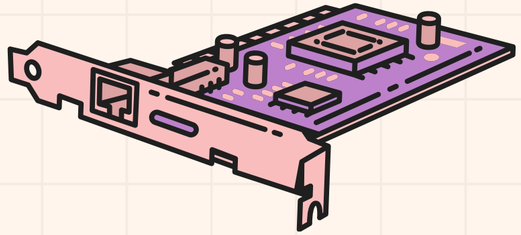
PROXY SERVER

ONE ENTRYPOINT



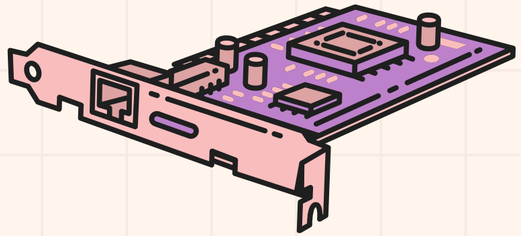
PROXY SERVER

ONE ENTRYPOINT



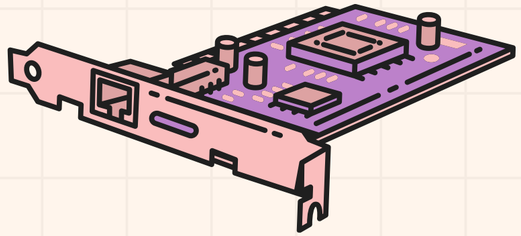
PROXY SERVER

ONE ENTRYPOINT



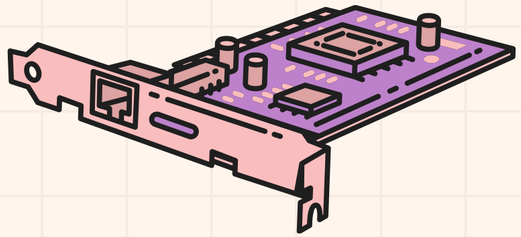
PROXY SERVER

ONE ENTRYPOINT



PROXY SERVER

ONE ENTRYPOINT



BENEFITS OF ONE ENTRYPOINT

HIGHER SECURITY

EASY MANAGEMENT

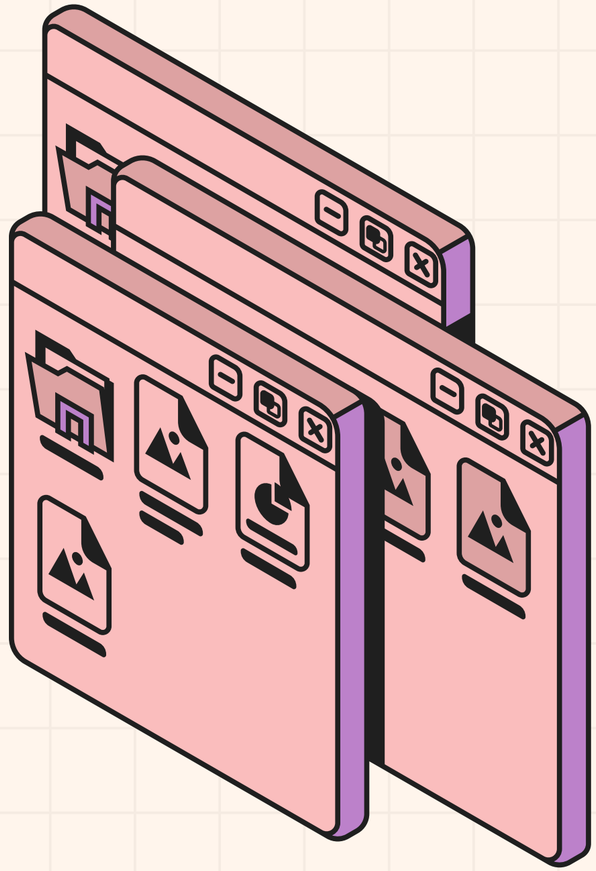
INCREASED FLEXIBILITY

VIRTUAL HOST IN NGINX

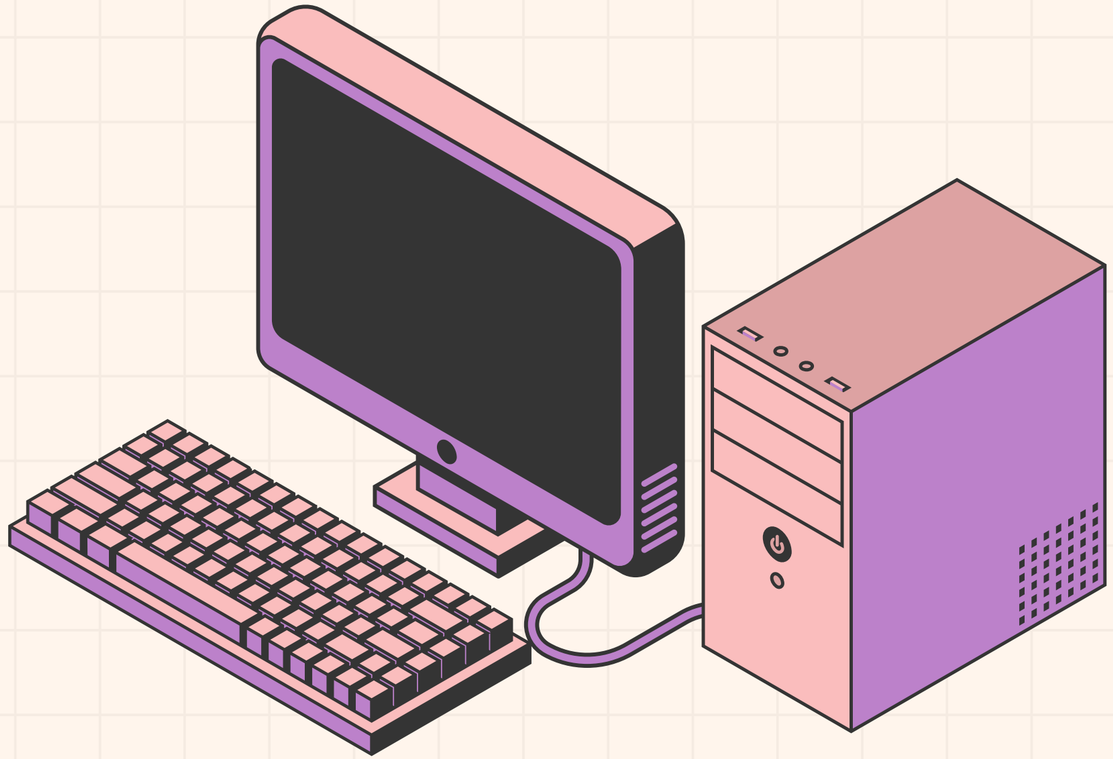
- In Nginx, virtual hosts are known as Server Blocks, allowing you to host multiple websites on a single server.

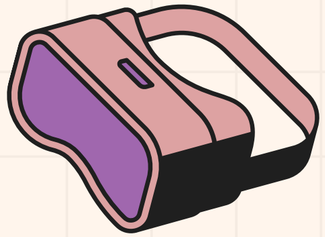
- This is achieved by configuring separate blocks within the **/etc/nginx/sites-available** directory, each with its own server_name (domain) and listen directive (port).

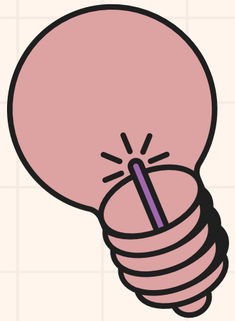
- Local testing: Configure **/etc/hosts** file.
  - Add 3 local domains mapping:
    - 127.0.0.1 demo1.test
    - 127.0.0.1 demo2.test
    - 127.0.0.1 demo3.test



DEMO







THANK YOU