Nginx

Thursday, December 19, 2024 9:17 PM

What is Nginx?

- Nginx is a powerful web server and uses a non-threaded,event-threaded architecture.
- It can also do other important things such as **load balancing** , and **HTTP caching** or used as **reverse proxy.**

What is Proxy Server?
A **proxy server** is an intermediary server that sits between a client (like your computer or device) and another server (like a website or service). It processes requests and responses, providing various functionalities depending on how it's configured.

Key Functions of a Proxy Server:

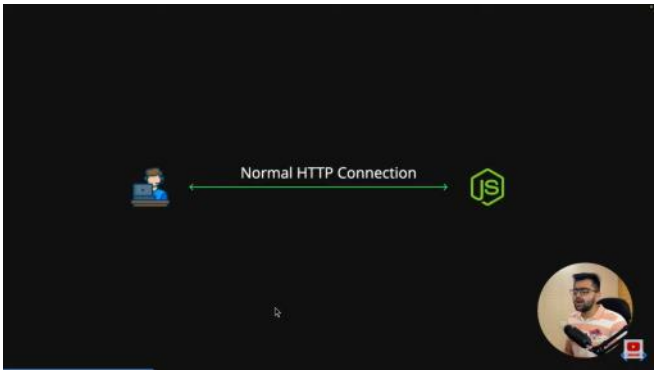
1. **Anonymity**: Hides the client's IP address, providing privacy by masking the user's location and identity.
2. **Access Control**: Blocks or allows access to specific websites or resources based on predefined rules.
3. **Caching**: Stores copies of frequently accessed resources (like websites) to improve speed and reduce bandwidth usage.
4. **Security**: Adds an additional layer of protection by filtering malicious content, blocking threats, and securing communications.
5. **Content Filtering**: Restricts access to certain content based on organizational policies (e.g., in schools or offices).
6. **Load Balancing**: Distributes incoming traffic across multiple servers to optimize performance and prevent overload.

How It Works:

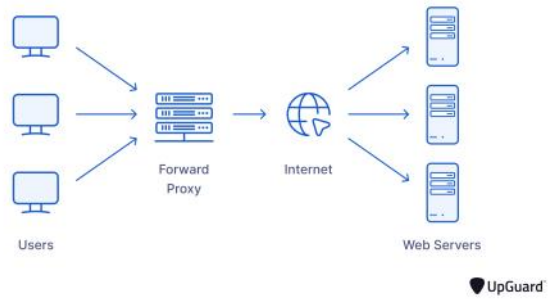
1. The client sends a request (e.g., accessing a webpage).
2. The proxy server receives the request, processes it, and forwards it to the target server.
3. The target server responds to the proxy server, which then relays the response back to the client.

Types of Proxy Servers:

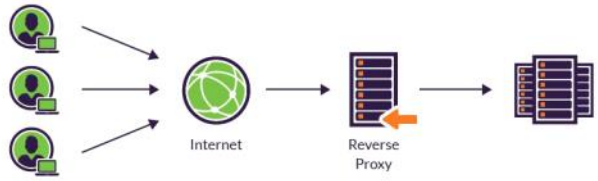
1. **Forward Proxy**: Acts on behalf of clients, often used for anonymity or bypassing restrictions.
2. **Reverse Proxy**: Sits in front of a server and handles incoming requests to enhance performance and security.
3. **Transparent Proxy**: Doesn't hide the client's IP address and is used primarily for caching and monitoring.
4. **Anonymous Proxy**: Hides the client's identity but reveals that a proxy is being used.
5. **High-Anonymity Proxy**: Completely conceals the client's identity and doesn't reveal proxy usage.



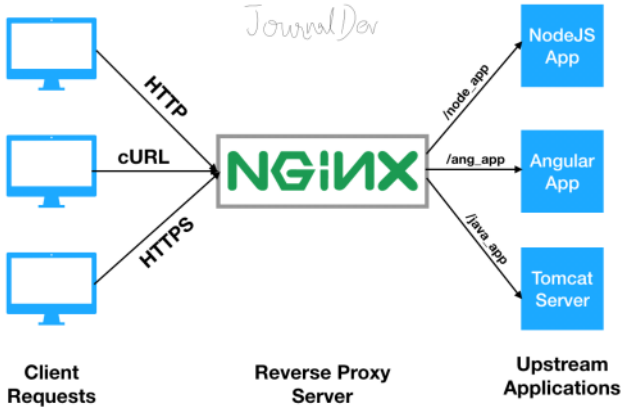
1. **Forward Proxy**: Acts on behalf of clients, often used for anonymity or bypassing restrictions.



2. **Reverse Proxy**: Sits in front of a server and handles incoming requests to enhance performance and security.

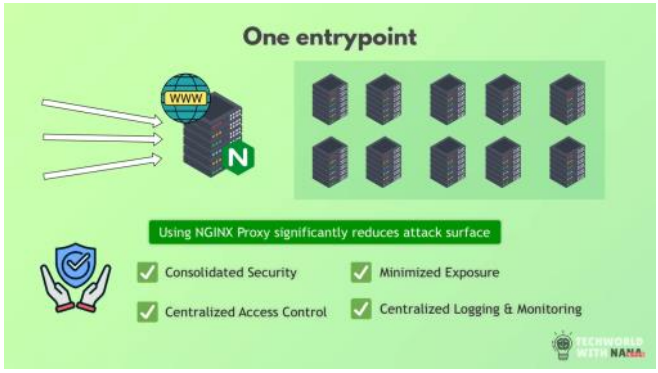


Basic Example Of Nginx:

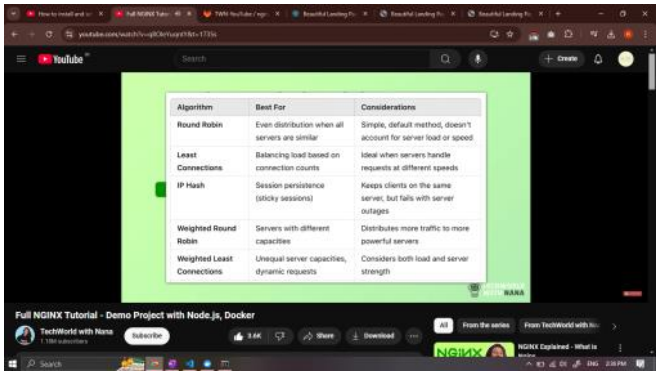


Nginx Provide :

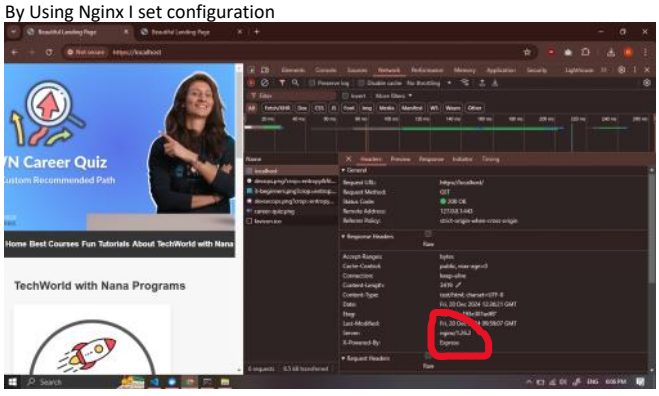
We can use set one port to nginx server



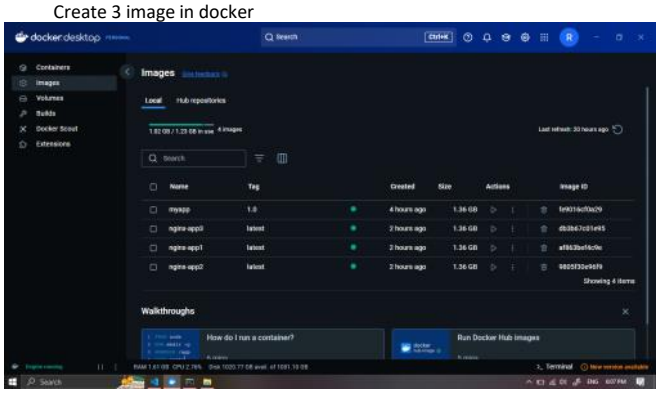
Different Algorithm for load balancing



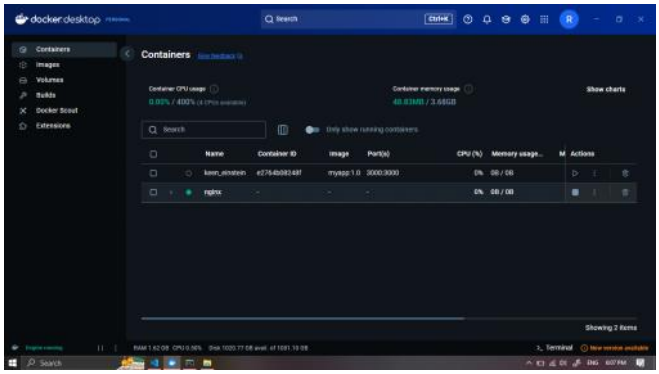
By Using Nginx I set configuration



Create 3 image in docker



Run container

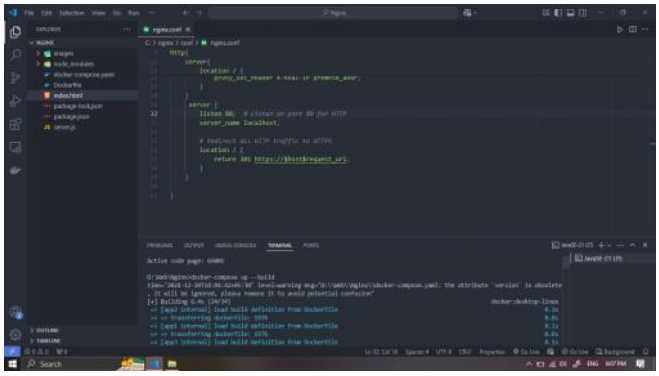


I Run All 3 Container

First Create Docker Image

Second create docker-compose.yaml file

docker-compose up --build

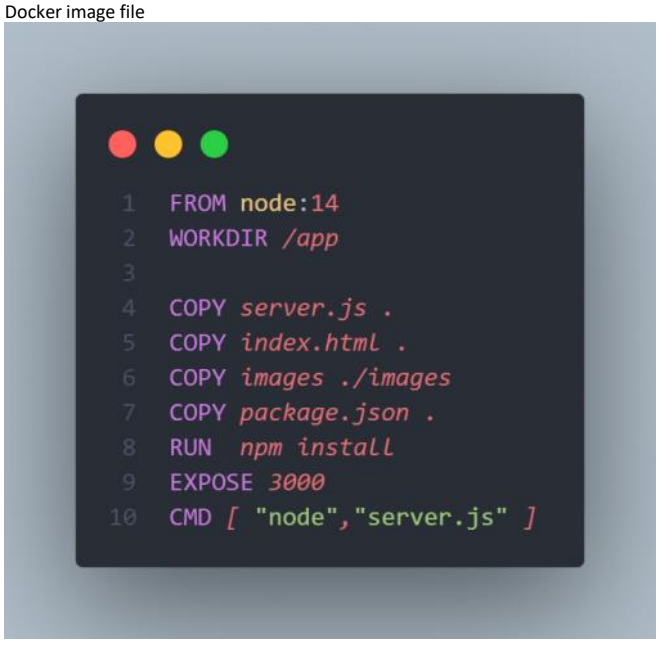


Create Simple Server

```
const express = require("express");
const path = require("path");

const app = express();
const port = 3000;

const replicaApp = process.env.APP_NAME;

app.use("/images", express.static(path.join(__dirname, "images")));

app.use("/", (req, res) => {
  res.sendFile(path.join(__dirname, "index.html"));
  console.log(`Request Served By ${replicaApp}`);
});

app.listen(port, () => {
  console.log(`${replicaApp} Server Started`);
});
```

Index.html file

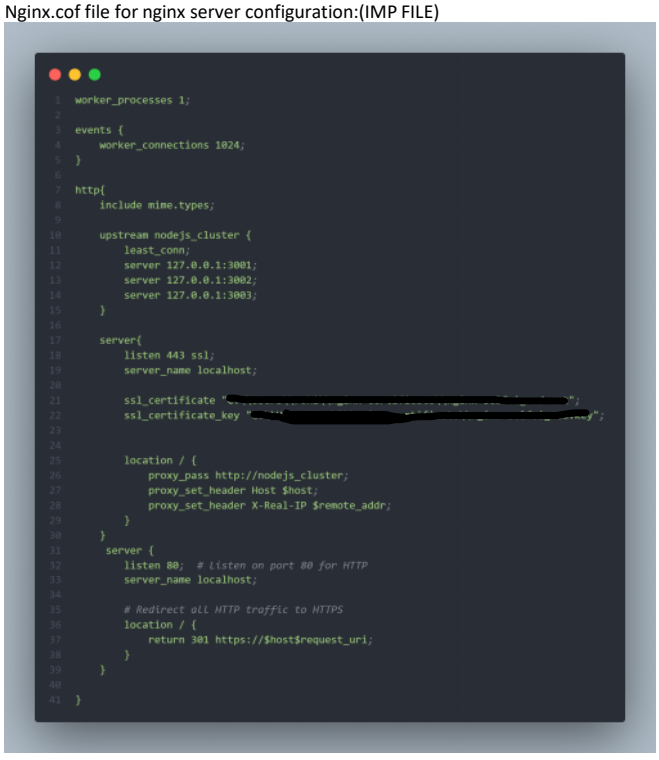
Docker image file



Docker-compose.yaml file

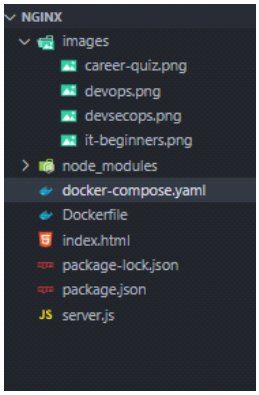
```yaml
version: '3'
services:
  app1:
    build: .
    environment:
      - APP_NAME=app1
    ports:
      - "3001:3000"
  app2:
    build: .
    environment:
      - APP_NAME=app2
    ports:
      - "3002:3000"
  app3:
    build: .
    environment:
      - APP_NAME=app3
    ports:
      - "3003:3000"
```

Nginx.cof file for nginx server configuration:(IMP FILE)



```
worker_processes 1;

events {
    worker_connections 1024;
}

http{
    include mime.types;

    upstream nodejs_cluster {
        least_conn;
        server 127.0.0.1:3001;
        server 127.0.0.1:3002;
        server 127.0.0.1:3003;
    }

    server{
        listen 443 ssl;
        server_name localhost;

        ssl_certificate "[redacted]";
        ssl_certificate_key "[redacted]";


        location / {
            proxy_pass http://nodejs_cluster;
            proxy_set_header Host $host;
            proxy_set_header X-Real-IP $remote_addr;
        }
    }
     server {
        listen 80;  # Listen on port 80 for HTTP
        server_name localhost;

        # Redirect all HTTP traffic to HTTPS
        location / {
            return 301 https://$host$request_uri;
        }
    }

}
```

File Structure


NGINX
images
career-quiz.png
devops.png
devsecops.png
it-beginners.png
node_modules
docker-compose.yaml
Dockerfile
index.html
package-lock.json
package.json
server.js