

Configuration of Nginx

1. Created docker-compose.yml file as per the official documentation.

```
version: '3.8'
services:
  app:
    image: 'jc21/nginx-proxy-manager:latest'
    restart: unless-stopped
    ports:
      - '80:80'
      - '81:81'
      - '443:443'
    volumes:
      - './data:/data'
      - './letsencrypt:/etc/letsencrypt'
```

2. Run docker-compose up -d
3. Logged in to admin ui
Localhost:81 is the port
Logged in with default admin credential.
Email: admin@example.com
Password: changeme
After logged in created user with own username and password.
4. In the Edit proxy host pop-up in Nginx Proxy manager. It was showing to Enter Domain-names, Scheme(http/https), Forward hostname/ip, Forward port

My local react app was running on `http://localhost:3000`, I wanted to configure localhost React application to run on “<My_Name>.localhost”

So, I have added below data

Domain-names: chiragboghara.localhost

Scheme: http

Forward hostname: todo-redux-dev (my service in docker-compose file)

Forward port: 3000

5. Now I can access my application by navigating to `http://chiragboghara.localhost` in my web browser.

Configuration of Traefik

1. Created docker-compose.yml file as per the official documentation, used traefik image.

```
version: "3"
services:
  reverse-proxy:
    image: traefik:v2.11
    command: --api.insecure=true --providers.docker
    ports:
      - "80:80"
```

```

    - "8080:8080"
  volumes:
    - /var/run/docker.sock:/var/run/docker.sock
  todo-redux-dev:
    container_name: todo-react
    image: todo-frontend:1.0
    build:
      context: .
      dockerfile: Dockerfile
    ports:
      - "3000:3000"
    env_file:
      - .env
    labels:
      - traefik.http.routers.todo-redux
      dev.rule=Host(`chiragboghara.localhost`)

```

2. docker-compose up -d
3. head over to <http://localhost:8080/> to see admin dashboard.
4. Attached label to container (in my docker compose file).
5. Accessed <http://chiragboghara.localhost> in my web browser.

Configuration of Caddy

1. Created docker-compose file with caddy image from docker-hub.

```

version: "3.8"
services:
  caddy:
    image: caddy:latest
    ports:
      - "80:80"
      - "443:443"
    volumes:
      - ./Caddyfile:/etc/caddy/Caddyfile
  todo-redux-dev:
    container_name: todo-react
    image: todo-frontend:1.0
    build:
      context: .
      dockerfile: Dockerfile
    ports:
      - "3000:3000"
    env_file:
      - .env

```

2. Created Caddyfile in root, its content is as below.

```
chiragboghara.localhost {  
reverse_proxy todo-redux-dev:3000  
}
```

3. Run `docker-compose up -d`
4. Now, my React application is accessible at `chiragboghara.localhost`. Caddy is acting as a reverse proxy and forwarding requests to my React app running on port 3000 inside the Docker container.

Differences

Nginx:

Strengths: Powerful, widely used in big setups, great performance.

Considerations: Configuration can be complex for beginners, manual reloads.

Traefik:

Strengths: Tailored for dynamic environments (microservices, containers).

Considerations: May lack some advanced features, seen as lightweight.

Caddy:

Strengths: Emphasis on simplicity, automatic HTTPS, easy setup.

Considerations: Fewer advanced features might be limiting for big setups.

Choosing:

Nginx: For big, traditional setups with complex needs.

Traefik: Best for dynamic environments, containers, microservices.

Caddy: Quick setups, simplicity, and ease of use, especially in development.

Ultimately:

I think we should choose based on project needs – Nginx for big setups, Traefik for dynamic environments, and Caddy for simplicity and ease.