

Distributed Systems Extra Credit Lab

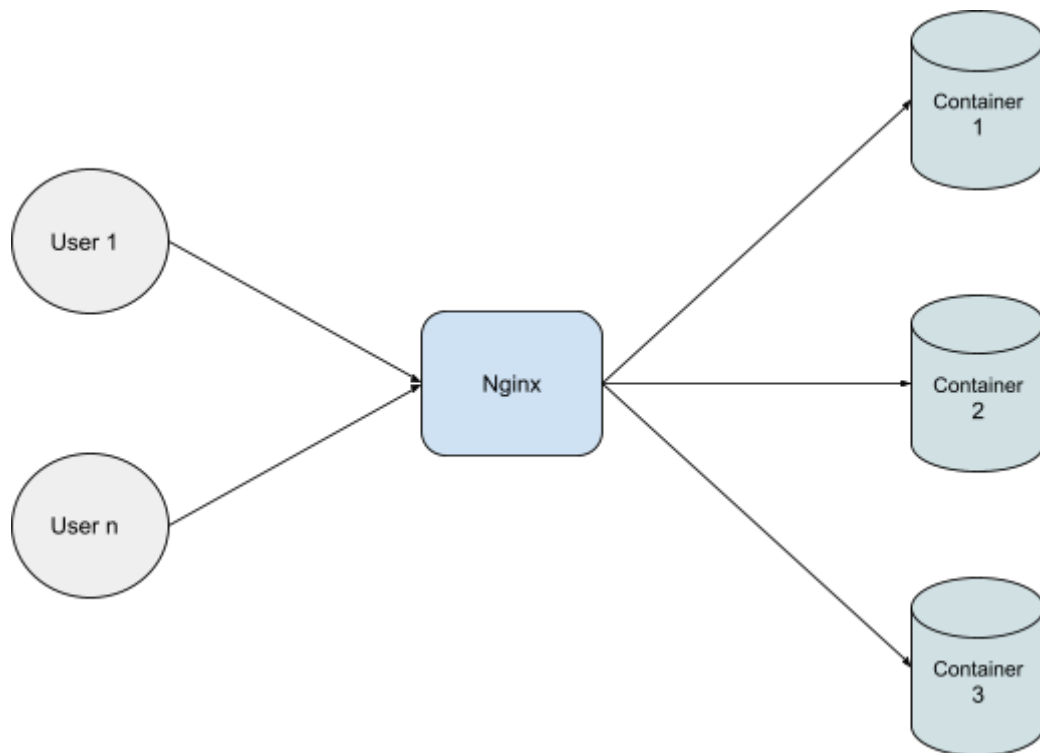
Load balancing Docker Containers using Nginx as a reverse proxy

By: Sartaj Bhuvaji

Motivation: Load balancers distribute incoming traffic across multiple servers, preventing any one server from becoming overwhelmed and ensuring website availability and reliability. Using load balancers can also improve website performance by optimizing resource utilization and reducing downtime.

Tech: Docker, Nginx

Architecture Diagram:



Description:

- Using Nginx, the user traffic can be distributed. But as there are multiple servers users must know every port number. This is done by Nginx as it acts as a reverse proxy and also load balances docker containers.
- We create a simple Flask application that has a download button to download a huge file.
- Docker Compose allows us to run multiple services for a single application. For the project, we have 3 docker applications that are running gunicorn and serve a flask website. We also have a single Nginx application.
- We let Nginx listen to all 'localhost' connections and forward the request to the server which has the minimum load.

Execution:

- Open Docker Desktop and log in.
 - Run the below commands in a terminal:
 - `$ Cd Dist\docker-load-balancer-using-nginx`
 - `$ docker-compose up -d --build --scale app=3`
 - Open the browser and go to “localhost”
 - Refresh the page to go to different docker containers.
 - Verify the Container ID displayed in the browser and Docker Desktop application.
-
- Open 1st “localhost” on one browser and click on Download.
 - By this, the docker container gets a small load and starts sending a file.
 - Quickly open “localhost” in a different browser and you can see that a different container ID is displayed.
 - You can also view the CPU load of a container in the Docker Desktop application.

Thus indicating Nginx is load balancing.

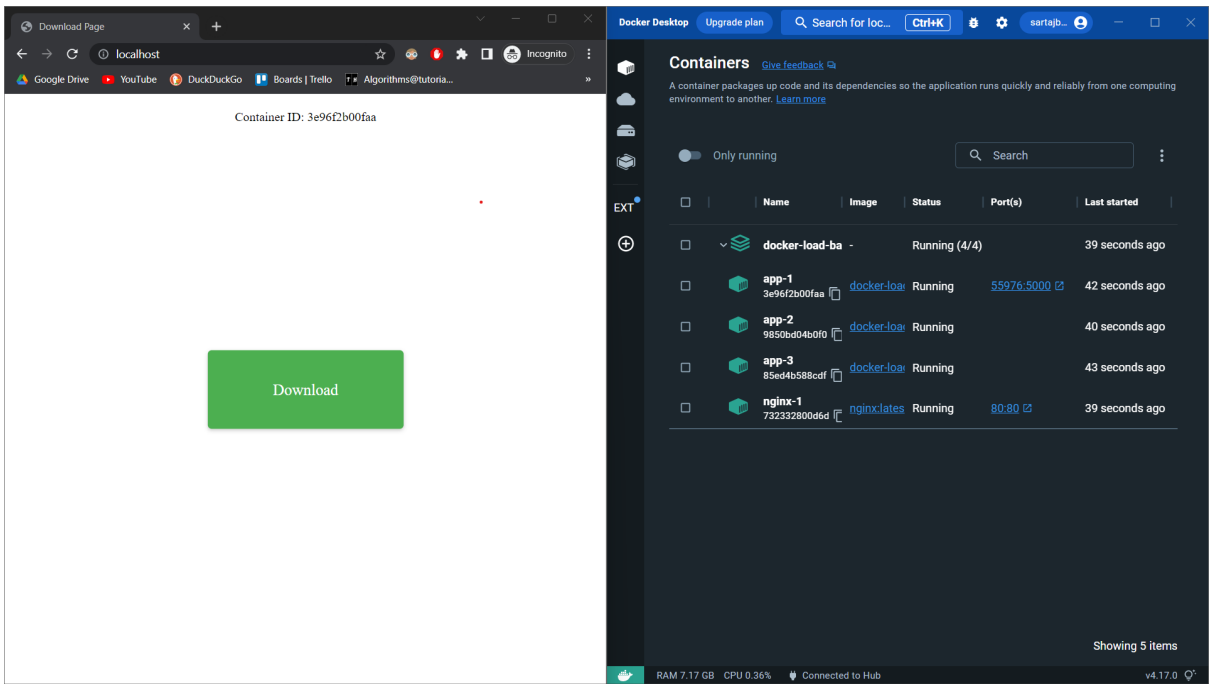
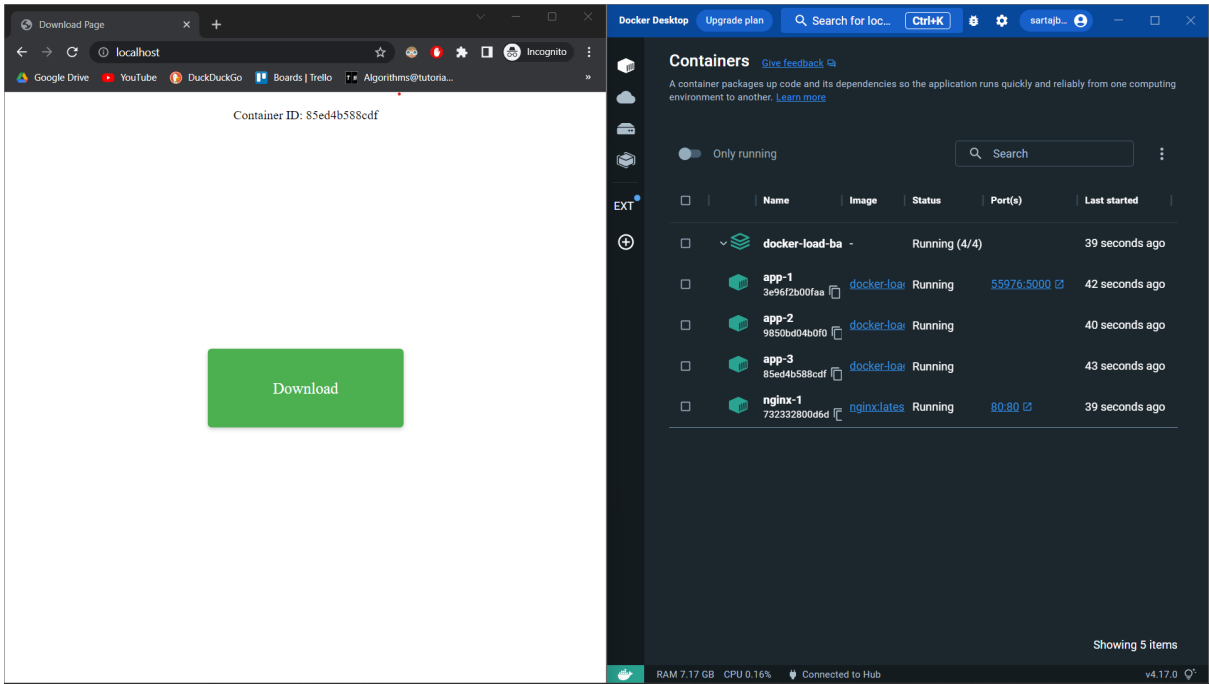
- To terminate docker containers run: `$ docker-compose down`

Output :

```
PS C:\Users\sbhuv\Desktop\Dist\docker-load-balancer-using-nginx> docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
693f90e33c74	nginx:latest	"/docker-entrypoint..."	7 minutes ago	Up 7 minutes
0.0.0.0:80->80/tcp	docker-load-balancer-using-nginx-nginx-1			
8af7f6739b3a	docker-load-balancer-using-nginx-app	"/bin/sh -c 'gunicorn..."	7 minutes ago	Up 7 minutes
0.0.0.0:54721->5000/tcp	docker-load-balancer-using-nginx-app-2			
1a8c4bce58d8	docker-load-balancer-using-nginx-app	"/bin/sh -c 'gunicorn..."	7 minutes ago	Up 7 minutes
0.0.0.0:54723->5000/tcp	docker-load-balancer-using-nginx-app-1			
59f0b3e0b64a	docker-load-balancer-using-nginx-app	"/bin/sh -c 'gunicorn..."	7 minutes ago	Up 7 minutes
0.0.0.0:54724->5000/tcp	docker-load-balancer-using-nginx-app-3			

Screenshots :



- In the below screenshot, we can see that as the CPU utilization of a container is high the second browser is load balanced to another container with a different container ID.

