# Load Balancing using Docker Containers

## Overview

Docker containers are a popular technology for creating running instances of images. They can be used to hold web servers, utility code, web services, database instances, etc. They are more lightweight than VMs and may pre-configured instances are already available (alpine, nginx, etc).

Load balancing is a real world issue that addresses server load by exposing a single web address on the internet and routing it to a collection of servers. Several different algorithms can be used to select which server to forward a request to (e.g. round robin, least connections, least response time, etc).

In order to verify the implementation of the load balancer, it is necessary to be able to start several web servers and a multitude of clients. Docker is a good tool to provide this capability.

The goal of this project is twofold. The first part is to design and implement a load balancer. The second part is to learn Docker to be able to spin up instances of servers and clients. How many instances can be served up on each Docker computer?

## Load Balancers

Load balancers can work at the TCP, IP, FTP, UDP level or at the application level (e.g. HTTP).

The first task to accomplish is to understand load balancers ad review the various algorithms that are typically used. The project could be limited to implementing 1 type of algorithm or several algorithms depending on the bandwidth of the team.

The design should support 2 – n servers. For this project, assume that all of the servers provide identical content. Any request from the external clients can be forwarded to any server.

## Docker

Docker can be used to create a container to run code. It is typically a Linux image and you can use standard package tools to add the specific packages that you need for your implementation.

It is possible to mount part of the host computers file system into the container image so that it can be accessed at run time.

Docker can also create multiple instances from a single container definition.

### Web Servers

Choose a web server technology that had a Docker definition already defined. Most of these are stored on Docker Hub. (<https://hub.docker.com/>) You can install them using the docker command line tool. The Docker tutorials use NGINX, but other web servers are available.

Each instance of the web server should provide the same content. You can implement this by having them all access a network share or you could mount the same content from the host file system into each container.

## Clients

It is possible to write a complex client application or find a client that does automatic requests, but the easiest solution is probably to spin up a small Linux image and use Bash or Python scripts to execute curl commands in a repetitive manner.

It would be good to have different classes of clients. Some that pull primarily text (quick and fast), some that pull images (slower and more data) and some that do both.

## Measurements

The implementation will need to log the number of requests per client and any errors that occurred during the requests. It also needs to log the number of requests received by each server.

The analysis of this information should provide insight into the effectiveness of each algorithm.

The time stamps of each request and the length of time to get a response should also be measured. The number of clients and servers can be steadily increased to see when the docker containers run our of compute power on each computer running the containers. At what number of instances does the docker platform start to slow down.

## Suggestions

You can run the Docker app with a different number of processors. In order to simulate extra stress and latency, it would probably be best to limit the load balancer and web servers to 1 processor and give the client images more memory and processors. Load balancers are most useful when there is a lot of traffic and the backend servers can be overloaded.

## Useful Information

### Load Balancers

<https://www.nginx.com/resources/glossary/load-balancing/>

<https://www.liquidweb.com/kb/understanding-load-balancing/>

<https://en.wikipedia.org/wiki/Load_balancing_(computing>)

<https://f5.com/resources/white-papers/load-balancing-101-nuts-and-bolts>

<https://www.inlab.de/articles/free-and-open-source-load-balancing-software-and-projects.html>

### Docker

<http://unitstep.net/blog/2015/12/21/accessing-the-host-filesystem-from-a-docker-container-on-os-x-or-windows/>

<https://docs.docker.com/engine/tutorials/dockervolumes/>

<https://bobcares.com/blog/apache-docker/>

<https://docs.docker.com/engine/tutorials/dockerimages/>

<https://access.redhat.com/articles/1328953>

## Alternatives

Depending on the bandwidth and experience of the students, an alternative would be to review open source load balancers and evaluate them using these same techniques.