Download DietPi from \url{http://fuzon.co.uk/phpbb/viewtopic.php?f=8&t=9}

Install DietPi on sd card

In the installation screen you can choose your own software. For this project we need ftp, so make sure that the ftp is selected.

In the installation it will ask if you need the standalone version or another one. choose the standalone version.

To access the raspberry Pi's from your pc or laptop you can use ssh. In this project a special program called putty is used.

After you have installed dietPi on all the Raspberry Pi's you can stress test them to make sure that they are operating.

For this use install stress. The Dietpi does not make use of sudo since this program is also removed and you are always operating from the root. So you have to be really sure what you are doing. To install stress type: \newline

install stress \newline

To run the test type: \newline

stress --cpu 5 --io 5 --vm 5 --vm-bytes 10M --timeout 3600s \newline

Now the webserver will be installed on all Pi's. For this you can type: \newline

NGINX install \newline

Download \& unpack latest nginx-rtmp \newline

Download \& unpack NGINX \newline

Build nginx with nginx-rtmp \newline

install FFmpeg

edit the configuration file of NGINX so it can stream RTMP

start NGINX and you can stream. (test if this stream is working) \newline

The RTMP stream you are streaming has standard port 1935 for the stream. This is because of the protocol. \newline

After the RTMP stream is working the JW player can be installed. This has to be done in the folder where the web pages are. The JW player makes it possible to stream the video inside the browser. \newline

To see what files are on which Raspberry Cassandra is used to see where the files are. So after installing JW Player you can install Cassandra.

\subsubsection{Load balancer}

One Raspberry Pi 2 will be the load balancer. This can be done by adjusting the configuration file. To do this type: \newline

nano /etc/nginx/sites-available/default \newline

In the first line you can add the upstream module we need. To do this type: \newline

upstream backend \{ \newline

server backend1.example.com; \newline

server backend2.example.com; \newline

server backend3.example.com; \newline

\} \newline

The backend1, backend2 and backend3 can be ip addresses or internet addresses. I used the ip adresses of the other Pi's.

To make it operating we need to adjust another module that is inside the server part where somewhere stands: \newline

location / \{ \newline

... \newline

\} \newline

we need to adjust what is standing in the middle of it. There only needs to be the line: \newline

proxy\\_pass http://backend;

Now we have to restart the load balancer.

We now have our own load balancer.

\subsection{Laptop or Desktop}

In this research a Laptop was used to run some testing software. This could also be a normal desktop. The Laptop is running on the operating system Microsoft Windows 8.Apache JMeter is used to test if the server is streaming correctly \cite{jmeter}. \newline

A website or media player is needed to test if the the RTMP stream coming from your Raspberry Pi 2 is working.