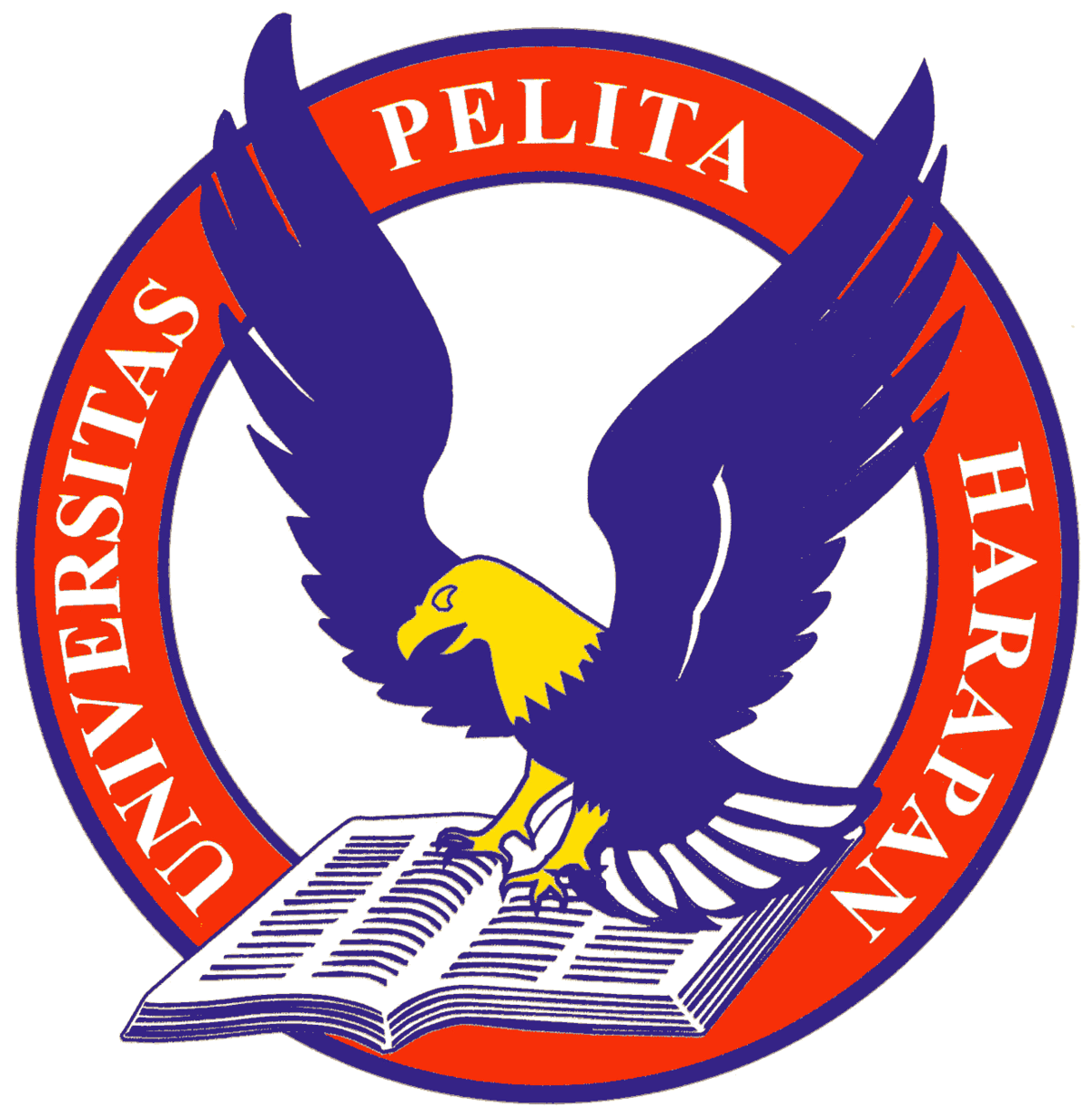
**Cloud Computing Report**



**Created By**

**Steven [00000011580] (https://github.com/NamaAkuSteven)**

**Jason [00000013271]**

**(https://github.com/XtraJos)**

**Table of content**

Table of Content 2

Introduction 3

Traffic 3

HaProxy 3

Client-Server 4

Peer-to-peer 4

About the Project 5

AWS 5

Bluemix 5

Vagrant 6

The Website 6

Refernce 7

**Introduction**

There is a time when we are unsure about our overall website. We want to keep it stable and protected; and thus we are going to do ‘load balancing’. Load balancing, according to [1], refers to efficiently distributing incoming network traffic across a group of backend servers, also known as a server farm or server pool. Load balancer is one of an essential tools used in website to increase capacity (concurrent users) and reliability of applications [2]. In short, load balancer is a tool to increase your website efficiency and security by providing different server that the website is stored [3].

There are four main components of load balancer that we are going to discuss:

1. **Traffic**

We usually find the word ‘traffic’ in a road, such as traffic light or traffic jam. The term sometimes associates with a crucial situation. In here, however, traffic means something completely different. According to [4], traffic means the amount of data moving across a network at a given point of time. Traffic Load Balancer enables the user to distribute traffic among multiple servers [5]. TLB leverages the Modular Port Concentrator (MPC)’s inline functionality, based on an enhanced version of equal-cost multipath (ECMP). Enhanced ECMP facilitates the distribution of flows across groups of servers. Enhancements to native ECMP ensure that when servers fail, only flows associated with those servers are impacted, minimizing the overall network churn on services and sessions.

**2. HaProxy**

In order to understand HaProxy, first, we have to know about proxy. What is a proxy? Acording to [6], a proxy is a dedicated computer or a software system running on a computer that acts as an intermediary between an endpoint device, such as a computer, and another server from which a user or client is requesting a service. HaProxy (High Availability Proxy) on the other hand, is an open source software TCP/HTTP Load Balancer [7]. Its most common use is to improve the performance and reliability of a server environment by distributing the workload across multiple servers (e.g. web, application, and database). It is used in many high-profile environments, including: GitHub, Imgur, Instagram, and Twitter.

**3. Server client**

A client is a user; us, who use the computer. A server is a program. Client/server is a program relationship in which one program (the client) requests a service or resource from another program (the server). Although the client/server model can be used by programs within a single computer, it is a more important concept for networking.  In this case, the client establishes a connection to the server over a local area network (LAN) or wide-area network (WAN), such as the Internet. Once the server has fulfilled the client's request, the connection is terminated. Your Web browser is a client program that has requested a service from a server; in fact, the service and resource the server provided is the delivery of this Web page [8].

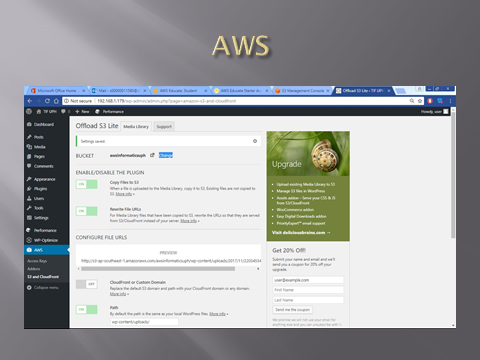
**4. Peer-to-peer**

The word ‘peer’ is such an underrated word for everyday live, but in computer science, we always use it. What is exactly is this word mean? Peer means a friend or a person with the same legal status. From that word alone, come the term peer-to-peer. Peer-to-peer is a connected two or more PCs that share resources without going through a separate server computer. A P2P network can be an ad hoc connection—a couple of computers connected via a Universal Serial Bus to transfer files [9].

**About the Project**

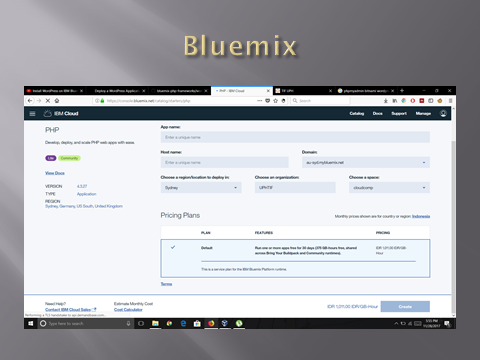
**AWS**

We have tried our best to use AWS in our website, but we find dead end. The dead end is where we tried to open AWS database website, but it says bucket is empty. As there is no time to continue AWS (And there are other project too), we finish AWS here.



Bluemix

We want to use Bluemix, but some of its component is limited and not free. So, we abandon the idea completely.

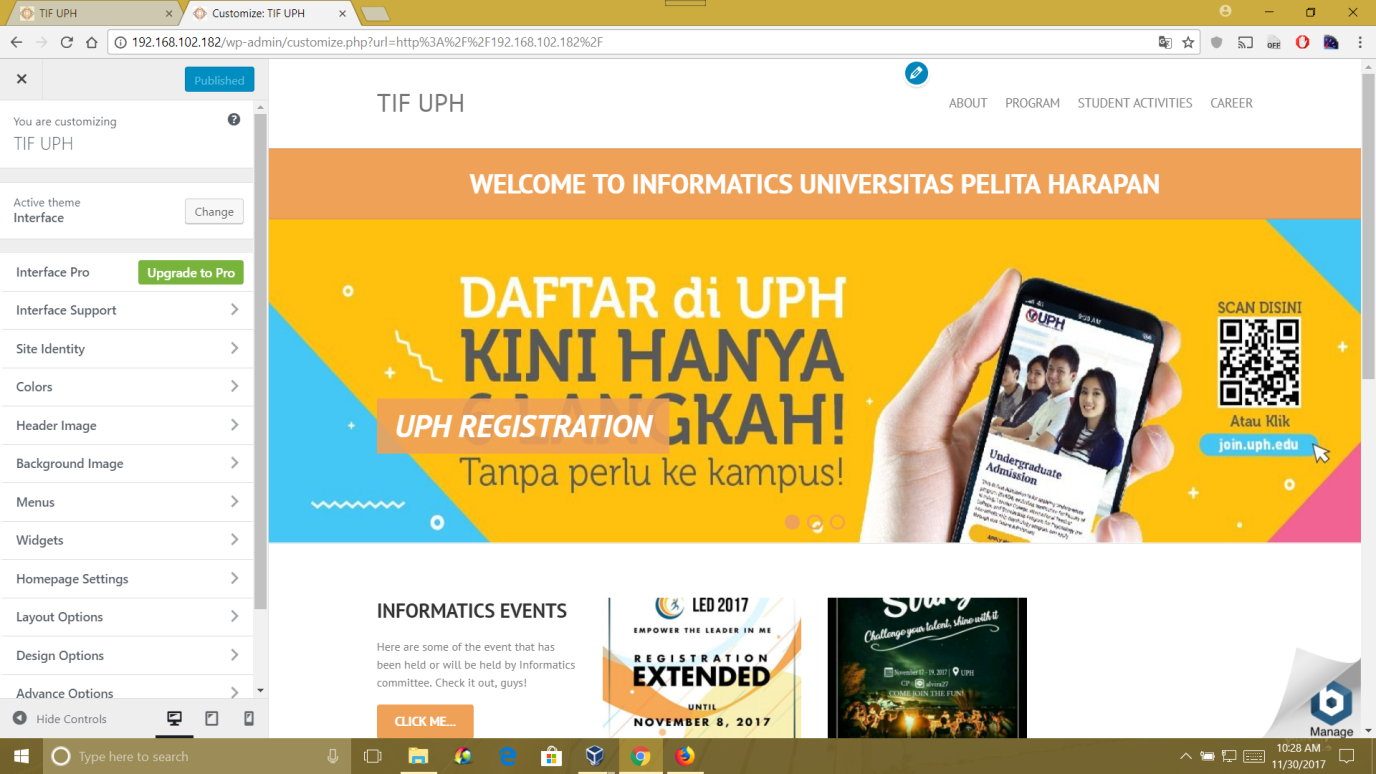


Vagrant

We have to tried tweaking our Vagrant code (which can be seen in the Github), but it also fails miserably. When we try to compile, it gives us error.



The Website



Reference

[1] – <https://www.nginx.com/resources/glossary/load-balancing/>

[2] – <https://f5.com/glossary/load-balancer>

[3] – <https://www.youtube.com/watch?v=zHn2G71hoIk>

[4] – <https://www.techopedia.com/definition/29917/network-traffic>

[5] – <https://www.juniper.net/documentation/en_US/junos/topics/concept/tdf-tlb-overview.html>

[6] – <http://whatis.techtarget.com/definition/proxy-server>

[7] – <https://www.digitalocean.com/community/tutorials/an-introduction-to-haproxy-and-load-balancing-concepts>

[8] – <http://searchnetworking.techtarget.com/definition/client-server>

[9] – <https://www.computerworld.com/article/2588287/networking/peer-to-peer-network.html>