### SERVIDORES WEB DE ALTAS PRESTACIONES

Christoforos Dellios

ID: AZ699940

(The exercises may be wrong or inaccurate due to the lack of understanding.I did my best )

EXERCISES

TEMA 1

EXERSICE 1 :Find information about the tasks or web services for which programs we discussed at the beginning of the session is more used:

**Apache**: Apache is the most widely used web server software. Developed and maintained by Apache Software Foundation, Apache is an open source software available for free. It runs on 67% of all webservers in the world. It is fast, reliable, and secure. It can be highly customized to meet the needs of many different environments by using extensions and modules.

**NGINX** is open source software for web serving, reverse proxying, caching, load balancing, media streaming, and more. It started out as a web server designed for maximum performance and stability. In addition to its HTTP server capabilities, NGINX can also function as a proxy server for email (IMAP, POP3, and SMTP) and a reverse proxy and load balancer for HTTP, TCP, and UDP servers.

**thttpd** (tiny/turbo/throttling HTTP server) is an open source software web server from ACME Laboratories, designed for simplicity, a small execution footprint and speed.

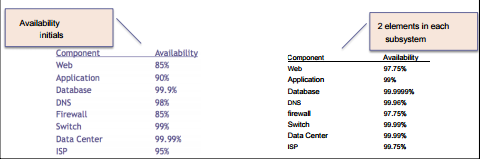
**Cherokee** is an open-source cross-platform web server that runs on Linux, BSD variants, Solaris, OS X, and Windows. It is a lightweight, high-performanceweb server/reverse proxy licensed under the GNU General Public License. Its goal is to be fast and fully functional yet still light. Major features of **Cherokee** include a graphical administration interface named *cherokee-admin*, and a modular light-weight design.

**Node**.**js** is a platform built on Chrome's **JavaScript**runtime for easily building fast and scalable network applications. **Node**.**js** uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.

TEMA 2

**EXERCISE 1**: Calculate system availability if we have two replicas of each element (total 3

elements in each subsystem).



We keep the first system as a data center and we use the formula for the other two replicas . So we have 3 components.

WEB: AS=85%+(1-85%)\*97,75%=99,66%+(1-99,66%)\*97,75%=99.992%

Application : As=90%+(1-90%)\*99%=99.9%+(1-99,9%)\*99%=99.999%

Database : As=99.9%+(1-99,9%)\*99,9999%=99,9999999%+(1-99,9999999%)\*99,9999%=99,9999999999999%

Dns: As=98%+(1-98%)\*99,96%=99.9992%+(1-99,9992%)\*99,96%=99,99999968%

Firewall : As=85%+(1-85%)\*97,75%=99.6625%+(1-99,6625)\*97,75%=99.99240625%

Switch: As=99%+(1-99%)\*99,99%=99.9999%+(1-99,9999)\*99,99%=99.99999999%

Data Center : As=99,99%+(1-99,99%)\*99,99%=99.999999%

ISP : As=95%+(1-95%)\*99,75%=99.9875%+(1-99,9875%)\*99,75%=99.99996875%

So the whole syste availability is : Web\*application\*database\*dns\*firewall\*switch\*data center \* isp =

99.992%\*99.999%\*99.9999999999999%\*99,99999968%\*99.99240625%\*99.99999999%\*99.999999%\*99.99996875%=99.98337%

**EXERCISE 2**: Find frameworks and libraries for different languages ​that allow applications to highly available with relative ease.

The frameworks and libraries I found (except from pm2 ) are :

Nodemon : Monitor for any changes in your node.js application and automatically restart the server

Naught: Zero downtime deployment for your Node.js server using builtin cluster API

Phusion Passenger :  web server and application server with support for Ruby, Python and Node.js. It is designed to integrate into the Apache Http or the nginx web server, but also has a mode for running standalone without an external web server.

MongoDB

**EXERCISE 3**: How to analyze the level of charge of each of the subsystems on the server?

Search tools and learn to use them. . . . Or remember how to use them!

In order to analyze the lever of charge of each subsystems on a server you need several analyze tools so you can monitor devices, services, ports or protocols and analyzing traffic on your server.This allows you to know exacltly if something goes wrongs .Some of these tools are:

Microsoft Network Monitor.

Microsoft Network Monitor is a packet analyzer that allows you to capture, view and analyze network traffic. This tool is handy for troubleshooting network problems and applications on the network. Main features include support for over 300 public and Microsoft proprietary protocols, simultaneous capture sessions, a Wireless Monitor Mode and sniffing of promiscuous mode traffic, amongst others.

# Performance Analysis of Logs (PAL) Tool

The PAL (Performance Analysis of Logs) tool reads in a performance monitor counter log (any known format) and analyzes it using complex, but known thresholds (provided). The tool generates an HTML based report that graphically charts important performance counters and throws alerts when thresholds are exceeded.

# SQL Server Agent Subsystems

A *subsystem* is a predefined object that represents a set of functionality available to a SQL Server Agent proxy. Each proxy has access to one or more subsystems. Subsystems provide security because they delimit access to the functionality that is available to a proxy. Each job step runs in the context of a proxy, except for Transact-SQL job steps. Transact-SQL job steps use the EXECUTE AS command to set the security context.

## Nagios

Nagios is a powerful network monitoring tool that helps you to ensure that your critical systems, applications and services are always up and running. It provides features such as alerting, event handling and reporting. The Nagios Core is the heart of the application that contains the core monitoring engine and a basic web UI. On top of the Nagios Core, you are able to implement plugins that will allow you to monitor services, applications, and metrics, a chosen frontend as well as add-ons for data visualisation, graphs, load distribution, and MySQL database support, amongst others.

**EXERCISE 4:** Find examples of software and hardware balancers (commercial products). Search commercial products for application servers. Search commercial products for storage servers.

Software balancers:

1. HaProxy – A TCP load balancer.
2. NGINX– A http load balancer with SSL termination support.
3. Mod\_athena – Apache based http load balancer.
4. Varnish – A reverse proxy based load balancer.
5. Balance – Open source TCP load balancer.
6. LVS – Linux virtual server offering layer 4 load balancing

Hardware balancers:

1. F5 BIG-IP load balancer
2. CISCO system catalyst
3. Barracuda load balancer
4. Coytepoint load balance