LTIC

Manual Técnico

GRUPO 004

Professores:

Ana Paula Afonso
Hugo Miranda

Equipa:

Anabela Rodrigues
Fernando Alberto
Marcio Domingues
Ricardo Sousa
Sergio Pinto
Yevgen Goncharuk

Maio / 2014

Índice

1. Introdução

O sistema de informação SAT integra diversos processos académicos e pode ser utilizado a diversos níveis. Um dos principais objetivos é a gestão e suporte a todas as atividades académicas relacionadas com os alunos, com foco nas inscrições, escolha e gestão de horários. Integra também funcionalidades de gestão de dados pessoais dos intervenientes (alunos e funcionários) e de gestão da informação relativa a disciplinas, cursos e departamentos.

Do ponto de vista da infraestrutura de rede, o projeto tem como principal objetivo garantir a fluidez da informação mesmo nas horas de maior trafego, proporcionando aos alunos, professores e serviços académicos da faculdade uma operação tranquila e sem falhas durante o processo de inscrições.

Neste manual técnico estão todos os comandos e configurações técnicas necessárias à reposição do funcionamento normal do sistema em caso de falha grave.

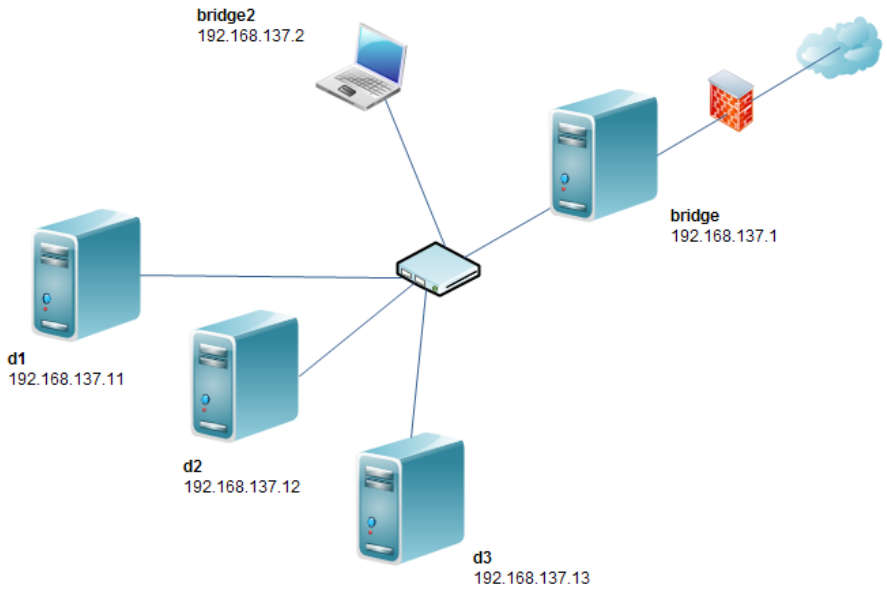
2. Hardware

O sistema principal é constituído por 5 máquinas, duas gateways (bridge e bridge2) e 3 réplicas (d1, d2, e d3).

2.1 Arquitetura lógica

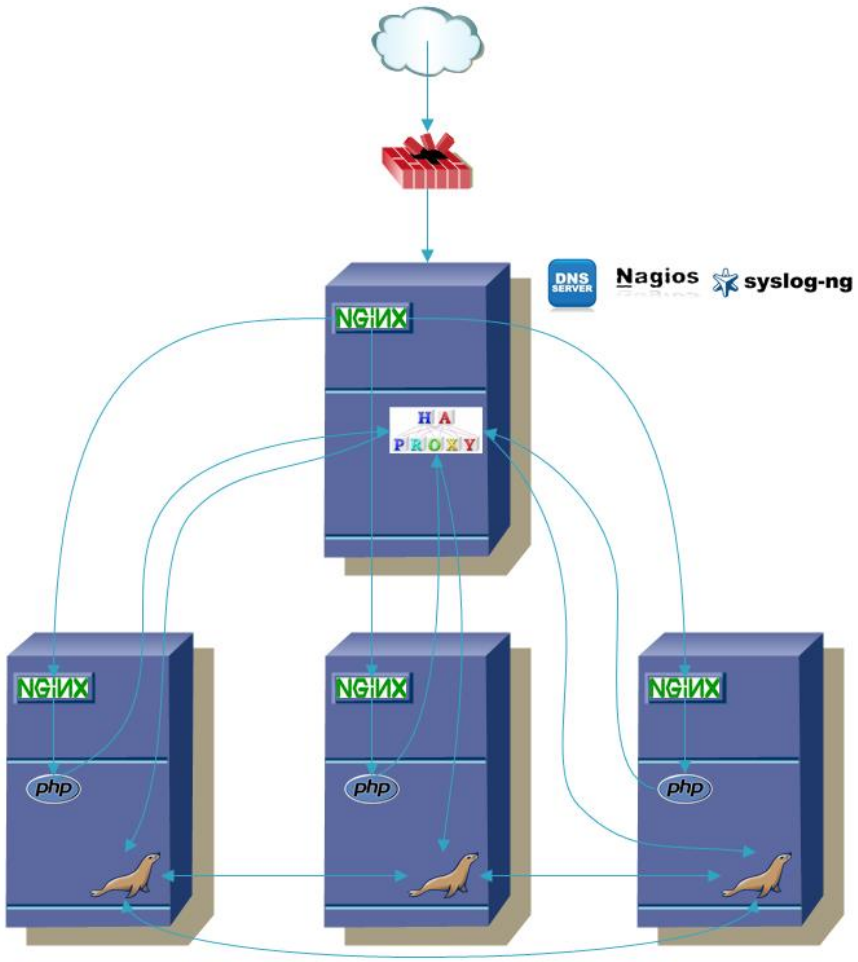
Hardware:
4 Servidores
1 portátil
1 Network Switch ligado a cada servidor por cabo de rede

Identificação dos servidores:
Bridge: 192.168.137.1
Bridge2: 192.168.137.2
d1: 192.168.137.11
d2: 192.168.137.12
d3: 192.168.137.13



2.2 Arquitetura física

As gateways recebem pedidos estáticos e dinâmicos. Os pedidos estáticos são imediatamente devolvidos pelo webserver NGINX ao cliente, enquanto que os dinâmicos são enviados para as réplicas recorrendo ao algoritmo Round Robin para avaliar qual a réplica disponível (balanceamento de carga). Quando o pedido chega à réplica escolhida, o webverver local NGINX usa o PHP para encaminhar o pedido, e no caso de usar a base de dados faz ligação para o servidor HaProxy para avaliar qual a réplica que vai processar o pedido (por exemplo uma leitura). Depois de escolhida a réplica e efetuado a operação, o galera cluster encarrega-se de replicá-la pelas restantes réplicas do sistema, e é então processado o commit e devolvida a informação ao webserver da gateway (bridge) que a fornece a resposta ao browser do cliente.



3. Réplicas (3)

Maquinas exatamente iguais tanto no harware como no sofware instalado (configurações incluídas). Responsáveis pelo armazenamento da informação geral do sistema (Webserver e Bases de dados). A base de dados instalada é a MariaDB e o cluster de replicação é o Galera Cluster.

Debian Server:
Download e instalação da versão mais recente em https://www.debian.org/distrib/

Criar Certificados:
O objetivo é substituir ferramentas inseguras como telnet, rlogin, etc. Permite a conexão de forma segura a todas as maquinas do sistema (a correr um servidor SSH). Mais rápido, mais prático e mais seguro.

cmd:
 # ssh-keygen

Importar Certificados para todas as réplicas:

cmd:
 # scp .ssh/rsa_key.pub user@d1:/tmp
 # scp .ssh/rsa_key.pub user@d2:/tmp
 # scp .ssh/rsa_key.pub user@d3:/tmp

Interfaces de rede:
A configuração da interface de rede é necessária para a máquina poder comunicar com as outras máquinas do sistema SAT e ter acesso à internet para fazer download dos diferentes módulos necessários ao funcionamento do sistema, bem como updates do SO.

Nota: O acesso à rede internet é feito a partir da maquina **bridge** com recurso a um interface de rede wireless usb cuja configuração se apresenta no ponto 4.

Ficheiro de configuração: (*/etc/network/interfaces*)

 # This file describes the network interfaces available on your system
 # and how to activate them. For more information, see interfaces(5).

 # The loopback network interface
auto lo
iface lo inet loopback

 # The primary network interface
allow-hotplug eth0
#iface eth0 inet dhcp
iface eth0 inet static
address 192.168.137.11
netmask 255.255.255.0
network 192.168.137.0
broadcast 192.168.137.255
gateway 192.168.137.1
dns-nameservers 8.8.8.8

Atualizar servidor:
Instalação dos updates do Debian. Essencial para a performance e segurança do sistema.

cmd:
 # sudo apt-get update
 # sudo apt-get upgrade

Hosts:
O ficheiro hosts faz o relacionamento entre um nome de computador e um endereço IP. Torna a operação e configuração do sistema mais prática e rápida.

Ficheiro de configuração: (*/etc/hosts)*

127.0.0.1 localhost

192.168.137.11 d1
192.168.137.12 d2
192.168.137.13 d3
 # The following lines are desirable for IPv6 capable hosts
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

NGINX Server:

Nginx é um webserver open source, um reverse proxy server para protocolos HTTP, SMTP, POP3 e IMAP e um application load balancer com um forte foco na performance e no baixo uso de memória e com inúmeras possibilidades de configuração para melhor performance.

Tecnicamente, o Nginx consome menos memória que o Apache, pois lida com requisições Web através do conceito de "event-based web server" (asynchronous) , já o Apache é baseado no conceito "process-based server" utilizando threads para tratar dos pedidos.

Nas réplicas faz a gestão dos pedidos dinâmicos à base de dados através do PHP.

Nota: Nas Gateways faz o balanceamento de carga distribuindo os pedidos pelas 3 réplicas, encaminha os pedidos da mesma sessão para a mesma réplica e ainda, faz a gestão dos pedidos com conteúdos estáticos e dinâmicos.

cmd:
 # apt-get install nginx

Ficheiro de configuração: (/etc/nginx/sites-enabled/default)

```
server {
#     location /static/ {
#           root /var/www/;
#     }
      location / {
            root /var/www;
      }

      location /aluno {
            location ~ \.php$ {
                  autoindex on;
                  try_files $uri $uri/ /index.php;
                  root /var/www;
                  fastcgi_split_path_info ^(.+\.php)(/.+)$;
                  fastcgi_pass unix:/var/run/php5-fpm-aluno.sock;
                  fastcgi_index index.php;
                  include fastcgi_params;
            }
      }
      location /professor {
            location ~ \.php$ {
                  autoindex on;
                  try_files $uri $uri/ /index.php;
```

```
            root /var/www;
            fastcgi_split_path_info ^(.+\.php)(/.+)$;
            fastcgi_pass unix:/var/run/php5-fpm-professor.sock;
            fastcgi_index index.php;
            include fastcgi_params;
        }
    }
    location /funcionario {
        location ~ \.php$ {
            autoindex on;
            try_files $uri $uri/ /index.php;
            root /var/www;
            fastcgi_split_path_info ^(.+\.php)(/.+)$;
            fastcgi_pass unix:/var/run/php5-fpm-funcionario.sock;
            fastcgi_index index.php;
            include fastcgi_params;
        }
    }
    location ~ \.php$ {
        autoindex on;
        try_files $uri $uri/ /index.php;
        root /var/www;
        fastcgi_split_path_info ^(.+\.php)(/.+)$;
        fastcgi_pass unix:/var/run/php5-fpm.sock;
        fastcgi_index index.php;
        include fastcgi_params;
    }
    location ~ /\.ht {
        deny all;
    }
}
```

PHP 5 / PHP-FPM:

PHP é uma linguagem open source muito comum em servidores web para produzir páginas dinâmicas.
Necessário ao servidor web NGINX no processamento dos pedidos dinâmicos à base de dados e na separação de privilégios dos utilizadores.

cmd:
 # apt-get install php5-fpm php5-mysql

Ficheiro de configuração: *(/etc/php5/fpm/pool.d/aluno.conf)*

```
[aluno]
listen = /var/run/php5-fpm-aluno.sock
listen.owner = www-data
listen.mode = 0600
user = aluno
#group = aluno
pm = dynamic
pm.max_children = 5
```

pm.min_spare_servers = 2
pm.max_spare_servers = 2

/etc/php5/fpm/pool.d/funcionario.conf
[funcionario]
listen = /var/run/php5-fpm-funcionario.sock
listen.owner = www-data
listen.mode = 0600
user = funcionario
#group =
pm = dynamic
pm.max_children = 5
pm.min_spare_servers = 2
pm.max_spare_servers = 2

/etc/php5/fpm/pool.d/professor.conf
[professor]
listen = /var/run/php5-fpm-professor.sock
listen.owner = www-data
listen.mode = 0600
user = professor
#group =
pm = dynamic
pm.max_children = 5
pm.min_spare_servers = 2
pm.max_spare_servers = 2

/etc/php5/fpm/pool.d/www.conf
[www]
user = www-data
group = www-data
listen = /var/run/php5-fpm.sock
pm = dynamic
pm.max_children = 5
pm.start_servers = 2
pm.min_spare_servers = 1
pm.max_spare_servers = 3
chdir = /

PhpMyAdmin:
PHPMyAdmin é uma ferramenta para administração do MySQL a partir de um web browser. Suporta uma vasta gama de operações em MySQL MariaDB e Drizzle. As operações mais frequentes são; administrar bases de dados, tabelas, colunas, inserir, remover e editar campos, utilizadores, permissões. Permite ainda executar código SQL.

cmd:
 # apt-get install phpmyadmin

Ficheiro de configuração: *(/etc/apt/sources.list)*

#
 # deb cdrom:[Debian GNU/Linux 7.4.0 _Wheezy_ - Official i386 NETINST Binary-1 20140208-12:25]/ wheezy main
#deb cdrom:[Debian GNU/Linux 7.4.0 _Wheezy_ - Official i386 NETINST Binary-1 20140208-12:25]/ wheezy main
deb http://ftp.pt.debian.org/debian/ wheezy main
deb-src http://ftp.pt.debian.org/debian/ wheezy main
deb http://ftp.pt.debian.org/debian/ wheezy non-free
deb-src http://ftp.pt.debian.org/debian/ wheezy non-free
deb http://security.debian.org/ wheezy/updates main
deb-src http://security.debian.org/ wheezy/updates main
 # wheezy-updates, previously known as 'volatile'
deb http://ftp.pt.debian.org/debian/ wheezy-updates main
deb-src http://ftp.pt.debian.org/debian/ wheezy-updates main
deb http://mirrors.fe.up.pt/pub/mariadb/repo/5.5-galera/debian wheezy main
deb-src http://mirrors.fe.up.pt/pub/mariadb/repo/5.5-galera/debian wheezy main

---

MariaDB + Galera Cluster:

MariaDB é uma base de dados open source que surgiu como fork do MySQL, criado pelo próprio fundador do projeto após sua aquisição pela Oracle. Esta base de dados é totalmente compatível com a MySQL e já está atualmente disponível nos repositórios de todas as distribuições de Linux.

O Galera Cluster fornece tecnologia assente em replicação síncrona multi-master. Significa que os dados estão mais seguros do que na replicação master-slave tradicional, porque estes são replicados imediatamente como parte do "commit". Além disso, na replicação "master-slave" tradicional, só pode efectuar "scale-out" para leitura, mas com MariaDB Galera Cluster pode ler e gravar em qualquer nó. Isso torna a vida mais fácil para os programadores, porque estes não precisam de separar transações de leitura e escrita, nas transações a submeter à BD.

Ideal para este sistema que requer mais leituras do que escritas.

Versão instalada: MariaDB 5.5 + Galera cluster 5.5

cmd:
 # apt-get install mariadb-galera-server galera

Ficheiro de configuração: *(/etc/mysql/conf.d/cluster.cnf)*

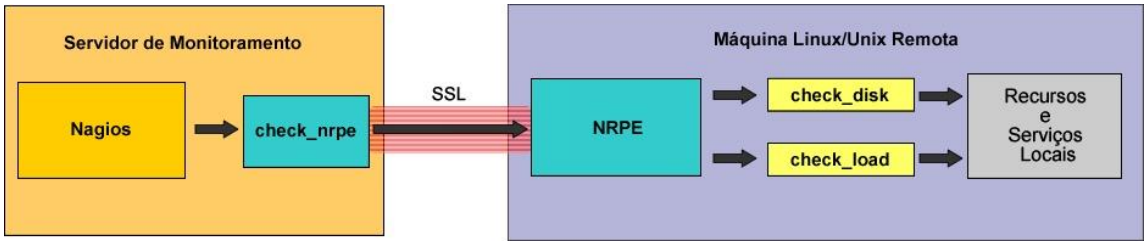
[mysqld]
query_cache_size=0
binlog_format=ROW
default_storage-engine=InnoDB
innodb_autoinc_lock_mode=2
query_cache_type=0
bind-address=0.0.0.0

wsrep_provider=/usr/lib/galera/libgalera_smm.so
wsrep_cluster_name="FTW"
wsrep_cluster_address="gcomm://192.168.137.11,192.168.137.12,192.168.137.13"
wsrep_sst_method=rsync

wsrep_node_address="192.168.137.11"
wsrep_node_name="d1"

Nagios Remote Plugin Executor (NRPE):
Nagios é uma aplicação open-source web-based que permitir monitorizar um número alargado de serviços (rede, servidores, hardware, etc) de um sistema informático em rede. Verifica em tempo real se estão ativos, emitindo alertas sempre que encontra algum problema.
http://www.nagios.org/

O **NRPE** é um agente que trabalha numa ou em várias máquinas remotas com o objetivo exclusivo de coletar informações e enviá-las ao servidor Nagios. A figura abaixo exemplifica o processo de comunicação entre o servidor Nagios e um cliente por meio do NRPE.



Em baixo a cinza estão as verificações configuradas para o sistema SAT.

cmd:
apt-get install nagios-plugins

Ficheiro de configuração: *(/etc/nagios/nrpe.cfg)*

log_facility=daemon
pid_file=/var/run/nagios/nrpe.pid
server_port=5666
nrpe_user=nagios
nrpe_group=nagios
allowed_hosts=127.0.0.1,192.168.137.1
dont_blame_nrpe=0
debug=0
command_timeout=60
connection_timeout=300
command[check_users]=/usr/lib/nagios/plugins/check_users -w 5 -c 10
command[check_load]=/usr/lib/nagios/plugins/check_load -w 15,10,5 -c 30,25,20
command[check_hda1]=/usr/lib/nagios/plugins/check_disk -w 20% -c 10% -p /dev/hda1
command[check_zombie_procs]=/usr/lib/nagios/plugins/check_procs -w 5 -c 10 -s Z
command[check_total_procs]=/usr/lib/nagios/plugins/check_procs -w 150 -c 200
include=/etc/nagios/nrpe_local.cfg
include_dir=/etc/nagios/nrpe.d/

service mysql start --wsrep-new-cluster #so correr na maquina que se ligar primeiro

RKHUNTER (Rootkit Hunter):
Scanner de deteção de malware para sistemas Unix/Linux.

cmd:
apt-get install rkhunter

Ficheiro de configuração: *(/etc/rkhunter.conf)*

ROTATE_MIRRORS=1
UPDATE_MIRRORS=1
MIRRORS_MODE=0
MAIL-ON-WARNING=""
MAIL_CMD=mail -s "[rkhunter] Warnings found for ${HOST_NAME}"
TMPDIR=/var/lib/rkhunter/tmp
DBDIR=/var/lib/rkhunter/db
SCRIPTDIR=/usr/share/rkhunter/scripts
UPDATE_LANG=""
LOGFILE=/var/log/rkhunter.log
APPEND_LOG=0
COPY_LOG_ON_ERROR=0
COLOR_SET2=0
AUTO_X_DETECT=1
WHITELISTED_IS_WHITE=0
ALLOW_SSH_ROOT_USER=no
ALLOW_SSH_PROT_V1=0
ENABLE_TESTS="all"
DISABLE_TESTS="suspscan hidden_procs deleted_files packet_cap_apps apps"
USER_FILEPROP_FILES_DIRS="/var/www/"*
SCRIPTWHITELIST=/bin/egrep
SCRIPTWHITELIST=/bin/fgrep
SCRIPTWHITELIST=/bin/which
SCRIPTWHITELIST=/usr/bin/groups
SCRIPTWHITELIST=/usr/bin/ldd
SCRIPTWHITELIST=/usr/bin/lwp-request
SCRIPTWHITELIST=/usr/sbin/adduser
SCRIPTWHITELIST=/usr/sbin/prelink
IMMUTABLE_SET=0
PHALANX2_DIRTEST=0
ALLOW_SYSLOG_REMOTE_LOGGING=1
SUSPSCAN_TEMP=/dev/shm
SUSPSCAN_MAXSIZE=10240000
SUSPSCAN_THRESH=200
USE_LOCKING=0
LOCK_TIMEOUT=300
SHOW_LOCK_MSGS=1
DISABLE_UNHIDE=1
INSTALLDIR="/usr"

RSYSLOG (rocket-fast system for log processing):
Aplicação open source usada em sistemas Unix/Linux para encaminhar logs em redes IP. Aceita inputs de várias fontes.
Neste sistema recebe os logs das máquinas locais, grava-os em /var/log/syslog e envia-os para a bridge (rsyslog server). Ficam disponíveis para serem consultados na interface web através do Loganalyser.

Ficheiro de configuração: *(/etc/rsyslog.d/remote.conf)*

```
/etc/rsyslog.d/remote.conf
$ModLoad imuxsock
$ModLoad imklog
*.*    @192.168.137.1:514
```

4. Bridge

Máquina responsável pela gestão do tráfego entre clientes e servidores. Assegura o balanceamento de carga pelas 3 réplicas, a gestão do DNS, a gestão da Firewall, a gestão do HaProxy e ainda do servidor Nágios.

Debian Server:
Download e instalação da versão mais recente em https://www.debian.org/distrib/

Rede Wireless:
O acesso à rede internet é feito a partir desta máquina com recurso a um interface de rede wireless usb configurado em bridge na rede Eduroam da FCUL. Esta máquina faz assim a ponte entre as redes da Fcul (Eduroam) e do sistema SAT através da porta de rede Ethernet, ligada ao switch do sistema por cabo de rede.

cmd:
apt-get install wpa-supplicant

Ficheiro de configuração 1: *(/etc/init.d/eduroam.sh)*

```
case "$1" in
    start)
        wpa_supplicant -i wlan0 -c /etc/wpa_supplicant/eduroam.conf &
        dhclient wlan0
        ;;
    stop)
        echo "WTF?"
        ;;
    *)
        echo "OI?!"
        exit 1
        ;;
esac
exit 0
```

Ficheiro de configuração 2: *(/etc/wpa_supplicant/eduroam.conf)*

```
network={
    ssid="eduroam"
    key_mgmt=WPA-EAP
    eap=TTLS
    identity="fc41176@alunos.fc.ul.pt"
    password="---"
    phase1="peaplabel=1"
    phase2="auth=MSCHAPV2"
}
```

Interfaces de rede:

A configuração da interface de rede é necessária para a máquina poder comunicar com as outras máquinas do sistema SAT e ter acesso à internet para fazer download dos diferentes módulos necessários ao funcionamento do sistema, bem como updates do SO.

Ficheiro de configuração: (*/etc/network/interfaces*)

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
    address 192.168.137.1
    netmask 255.255.255.0
    gateway 10.101.97.1
```

Nota: Após realizar a configuração, reiniciar o serviço de rede com:

```
# /etc/init.d/networking restart
```

Atualizar servidor:

Instalação dos updates do Debian. Essencial para a performance e segurança do sistema.

cmd:

```
sudo apt-get update
sudo apt-get upgrade
```

Hosts:

O ficheiro hosts faz o relacionamento entre um nome de computador e um endereço IP. Torna a operação e configuração do sistema mais simplificada e rápida.

Ficheiro de configuração: (/etc/hosts)

```
127.0.0.1       localhost
127.0.1.1       bridge
192.168.137.11  d1
192.168.137.12  d2
192.168.137.13  d3

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

NGINX Server + Sticky Module:

Nginx é um webserver open source, um reverse proxy server para protocolos HTTP, SMTP, POP3 e IMAP e um application load balancer com um forte foco na performance e no baixo uso de memória e com inúmeras possibilidades de configuração para melhor performance.

Tecnicamente, o Nginx consome menos memória que o Apache, pois lida com requisições Web através do conceito de "event-based web server" (asynchronous) , já o Apache é baseado no conceito "process-based server" utilizando threads para tratar dos pedidos.

O servidor Web Nginx garante o balanceamento de carga pelas 3 réplicas (utilizando o algoritmo Round Robin) nos pedidos para conteúdos dinâmicos (pedidos à bases de dados) e responde imediatamente aos pedidos de conteúdos estáticos.

O módulo adicional Sticky, garante que os pedidos dinâmicos de uma sessão sejam servidos sempre a partir da mesma réplica.

DOWNLOAD DO NGINX DO SITE OFFICIAL, DOWNLOAD STICKY MODULE (https://github.com/lusis/nginx-sticky-module)

cmd:

```
# apt-get install libssl-dev
# apt-get install zlib1g-dev
# apt-get install libpcre3-dev
# apt-get source nginx
./configure --without-mail_smtp_module --without-mail_pop3_module --without-mail_imap_module  --add-module=/home/user/nginx-sticky-module-1.1
```

Fcheiro de configuração: *(/etc/nginx/sites-enabled/default)*

```
upstream frontend{
    server d1:80;
    server d2:80;
    server d3:80;
```

```
    sticky;
}
server{
    location /static/ {
        root /var/www/;
    }
    location / {
        proxy_pass http://frontend;
    }
    location /RequestDenied {
        return 418;
    }
}
```

HaProxy:

HAProxy é uma aplicação gratuita, muito rápida e fiável. Oferece uma solução de load balancing e proxy para aplicações baseadas em TCP e HTTP distribuindo pedidos por múltiplos servidores.

Neste projeto é usado para fazer o balanceamento dos pedidos dinâmicos à base de dados replicada.

cmd:

apt-get install haproxy

Ficheiro de configuração: *(/etc/haproxy/haproxy.cfg)*

```
global
    log /dev/log    local0
    log /dev/log    local1 notice
    chroot /var/lib/haproxy
    user haproxy
    group haproxy
    daemon

defaults
    log     global
    mode tcp
    option  dontlognull
    retries 2
    contimeout 5000
    clitimeout 50000
    srvtimeout 50000

listen mariadb
    bind 0.0.0.0:3306
    mode tcp
    balance roundrobin
    option mysql-check user haproxy
    server d1 192.168.137.11 check fastinter 1000
    server d2 192.168.137.12 check fastinter 1000
```

server d3 192.168.137.13 check fastinter 1000

IpTables:

Netfilter é o firewall padrão embutido no kernel linux. A sua função é tratar regras aplicadas a pacotes TCP-IP. O iptables nada mais é que uma interface controladora do Netfilter. Atualmente é parte de todas as distribuições do Linux. Por defeito vem configurado para permitir todo o trafego.

Em baixo estão as regras de configuração aplicadas ao Sistema SAT.

Ficheiro de configuração: *(/etc/iptables.up.rules)*

```
*nat
-A POSTROUTING -o wlan0 -j MASQUERADE
COMMIT

*filter
#ignorar tudo para 127.0.0.1/8 que vem de if's que nao e o lo
-A INPUT -i lo -j ACCEPT
-A INPUT ! -i lo -d 127.0.0.1/8 -j REJECT

#aceitar traffego que ja tem conexao
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

#aceitar tudo o que vem daqui para fora
-A OUTPUT -j ACCEPT

-A INPUT -p tcp --dport 80 -j ACCEPT
-A INPUT -p tcp --dport 443 -j ACCEPT
-A INPUT -p tcp --dport 8080 -s 192.168.137.0/24 -j ACCEPT
-A INPUT -p tcp --dport 22 -j ACCEPT
-A INPUT -p udp --dport 53 -j ACCEPT
-A INPUT -p udp --dport 514 -j ACCEPT
-A INPUT -p tcp --dport 3306 -s 192.168.137.0/24 -j ACCEPT
#ipforward
-A FORWARD -i wlan0 -o eth0 -m state --state RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -i eth0 -o wlan0 -j ACCEPT

-A INPUT -j REJECT
-A FORWARD -j REJECT

COMMIT
```

```
/etc/network/if-pre-up.d/iptables
#!/bin/bash
/sbin/iptables-restore < /etc/iptables.up.rules
```

Snort:
Snort é um IDS (Intrusion Detection System) que tem como principal função o desenvolvimento de análises de tráfego em tempo real de forma automática. Para além de devolver alertas em caso de ataque, permite ao administrador processar os ajustes necessários na firewall em função da informação recolhida.

cmd:
apt-get install snort

Nagios Server:
Nagios é uma aplicação open-source web-based que permitir monitorizar um número alargado de serviços (rede, servidores, hardware, etc) de um sistema informático em rede. Verifica em tempo real se estão ativos, emitindo alertas sempre que encontra algum problema.
http://www.nagios.org/

cmd:
apt-get install php5 apache2 nagios

Ficheiro de configuração 1: *(/etc/nagios3/nagios.cfg)*

log_file=/var/log/nagios3/nagios.log
cfg_file=/etc/nagios3/commands.cfg
cfg_dir=/etc/nagios-plugins/config
cfg_dir=/etc/nagios3/conf.d
object_cache_file=/var/cache/nagios3/objects.cache
precached_object_file=/var/lib/nagios3/objects.precache
resource_file=/etc/nagios3/resource.cfg
status_file=/var/cache/nagios3/status.dat
status_update_interval=10
nagios_user=nagios
nagios_group=nagios
check_external_commands=1
command_check_interval=5s
command_file=/var/lib/nagios3/rw/nagios.cmd
external_command_buffer_slots=4096
lock_file=/var/run/nagios3/nagios3.pid
temp_file=/var/cache/nagios3/nagios.tmp
temp_path=/tmp
event_broker_options=-1
log_rotation_method=d
log_archive_path=/var/log/nagios3/archives
use_syslog=1
log_notifications=1
log_service_retries=1
log_host_retries=1
log_event_handlers=1
log_initial_states=0
log_external_commands=1
log_passive_checks=1
service_inter_check_delay_method=s

max_service_check_spread=30
service_interleave_factor=s
host_inter_check_delay_method=s
max_host_check_spread=30
max_concurrent_checks=0
check_result_reaper_frequency=10
max_check_result_reaper_time=30
check_result_path=/var/lib/nagios3/spool/checkresults
max_check_result_file_age=3600
cached_host_check_horizon=15
cached_service_check_horizon=15
enable_predictive_host_dependency_checks=1
enable_predictive_service_dependency_checks=1
soft_state_dependencies=0
auto_reschedule_checks=0
auto_rescheduling_interval=30
auto_rescheduling_window=180
sleep_time=0.25
service_check_timeout=60
host_check_timeout=30
event_handler_timeout=30
notification_timeout=30
ocsp_timeout=5
perfdata_timeout=5
retain_state_information=1
state_retention_file=/var/lib/nagios3/retention.dat
retention_update_interval=60
use_retained_program_state=1
use_retained_scheduling_info=1
retained_host_attribute_mask=0
retained_service_attribute_mask=0
retained_process_host_attribute_mask=0
retained_process_service_attribute_mask=0
retained_contact_host_attribute_mask=0
retained_contact_service_attribute_mask=0
interval_length=60
check_for_updates=1
bare_update_check=0
use_aggressive_host_checking=0
execute_service_checks=1
accept_passive_service_checks=1
execute_host_checks=1
accept_passive_host_checks=1
enable_notifications=1
enable_event_handlers=1
process_performance_data=0
obsess_over_services=0
obsess_over_hosts=0
translate_passive_host_checks=0
passive_host_checks_are_soft=0
check_for_orphaned_services=1
check_for_orphaned_hosts=1

```
check_service_freshness=1
service_freshness_check_interval=60
service_check_timeout_state=c
check_host_freshness=0
host_freshness_check_interval=60
additional_freshness_latency=15
enable_flap_detection=1
low_service_flap_threshold=5.0
high_service_flap_threshold=20.0
low_host_flap_threshold=5.0
high_host_flap_threshold=20.0
date_format=iso8601
p1_file=/usr/lib/nagios3/p1.pl
enable_embedded_perl=1
use_embedded_perl_implicitly=1
illegal_object_name_chars=`~!$%^&*|'"<>?,()=
illegal_macro_output_chars=`~$&|'"<>
use_regexp_matching=0
use_true_regexp_matching=0
admin_email=root@localhost
admin_pager=pageroot@localhost
daemon_dumps_core=0
use_large_installation_tweaks=0
enable_environment_macros=1
debug_level=0
debug_verbosity=1
debug_file=/var/log/nagios3/nagios.debug
max_debug_file_size=1000000
```

Ficheiro de configuração 2: *(/etc/nagios3/commands.cfg)*

```
define command{
        command_name    notify-host-by-email
        command_line    /usr/bin/printf "%b" "***** Nagios *****\n\nNotification Type:
$NOTIFICATIONTYPE$\nHost: $HOSTNAME$\nState: $HOSTSTATE$\nAddress:
$HOSTADDRESS$\nInfo: $HOSTOUTPUT$\n\nDate/Time: $LONGDATETIME$\n" | /usr/bin/mail
-s "** $NOTIFICATIONTYPE$ Host Alert: $HOSTNAME$ is $HOSTSTATE$ **" $CONTACTEMAIL$
        }
# 'notify-service-by-email' command definition
define command{
        command_name    notify-service-by-email
        command_line    /usr/bin/printf "%b" "***** Nagios *****\n\nNotification Type:
$NOTIFICATIONTYPE$\n\nService: $SERVICEDESC$\nHost: $HOSTALIAS$\nAddress:
$HOSTADDRESS$\nState: $SERVICESTATE$\n\nDate/Time: $LONGDATETIME$\n\nAdditional
Info:\n\n$SERVICEOUTPUT$\n" | /usr/bin/mail -s "** $NOTIFICATIONTYPE$ Service Alert:
$HOSTALIAS$/$SERVICEDESC$ is $SERVICESTATE$ **" $CONTACTEMAIL$
        }
define command{
        command_name    process-host-perfdata
        command_line    /usr/bin/printf "%b"
"$LASTHOSTCHECK$\t$HOSTNAME$\t$HOSTSTATE$\t$HOSTATTEMPT$\t$HOSTSTATETYPE$\t
```

```
$HOSTEXECUTIONTIME$\t$HOSTOUTPUT$\t$HOSTPERFDATA$\n" >> /var/lib/nagios3/host-perfdata.out
        }
define command{
        command_name    process-service-perfdata
        command_line    /usr/bin/printf "%b" "$LASTSERVICECHECK$\t$HOSTNAME$\t$SERVICEDESC$\t$SERVICESTATE$\t$SERVICEATTEMPT$\t$SERVICESTATETYPE$\t$SERVICEEXECUTIONTIME$\t$SERVICELATENCY$\t$SERVICEOUTPUT$\t$SERVICEPERFDATA$\n" >> /var/lib/nagios3/service-perfdata.out
        }
define command{
        command_name    check_maria
        command_line    /usr/lib/nagios/plugins/check_mysql -H d1 -u nagios -p nagiosISaPIMP! -P 3306
}
define command{
        command_name    check_replicas
        command_line    /usr/lib/nagios/plugins/check_replicas.sh
}
define command{
        command_name    check_google
        command_line    /usr/lib/nagios/plugins/check_ping -H 8.8.8.8 -w 3000.0,80% -c 5000,100%
}
define command{
        command_name    check_dns_local
        command_line    /usr/lib/nagios/plugins/check_dns -H 127.0.0.1
}
```

---

cat /usr/lib/nagios/plugins/check_replicas.sh
#script home made(BEST SCRIPT EVAH) para ver e avisar (error/warning) de quantidade de replicas de mariadb (3 para OK)
/usr/lib/nagios/plugins/check_mysql_query -H d1 -u nagios -p nagiosISaPIMP! -q "select VARIABLE_VALUE from information_schema.GLOBAL_STATUS where VARIABLE_NAME = 'wsrep_cluster_size';" |perl -wlne '$a=$1 if /(\d+)/; $b="OK $a" if $a eq 3; $b="WARNING $a" if $a eq 2; $b="CRITICAL $a" if $a eq 1; print $b; exit 1 if $a eq 2; exit 2 if $a eq 1;'

Ficheiro de configuração 3: *(/etc/nagios3/conf.d/localhost_nagios2.cfg)*

```
# A simple configuration file for monitoring the local host
# This can serve as an example for configuring other servers;
# Custom services specific to this host are added here, but services
# defined in nagios2-common_services.cfg may also apply.
#

define host{
        use             generic-host        ; Name of host template to use
        host_name           localhost
        alias           localhost
        address           127.0.0.1
```

```
    }

# Define a service to check the disk space of the root partition
# on the local machine.  Warning if < 20% free, critical if
# < 10% free space on partition.

define service{
    use                     generic-service       ; Name of service template to use
    host_name               localhost
    service_description     Disk Space
    check_command           check_all_disks!20%!10%
    }

# Define a service to check the number of currently logged in
# users on the local machine.  Warning if > 20 users, critical
# if > 50 users.

define service{
    use                     generic-service       ; Name of service template to use
    host_name               localhost
    service_description     Current Users
    check_command           check_users!20!50
    }

# Define a service to check the number of currently running procs
# on the local machine.  Warning if > 250 processes, critical if
# > 400 processes.

define service{
    use                     generic-service       ; Name of service template to use
    host_name               localhost
    service_description     Total Processes
        check_command           check_procs!250!400
    }

# Define a service to check the load on the local machine.

define service{
    use                     generic-service       ; Name of service template to use
    host_name               localhost
    service_description     Current Load
        check_command           check_load!5.0!4.0!3.0!10.0!6.0!4.0
    }

#define service{
#     use     generic-service
#     host_name      localhost
#     service_description    SSH
#          check_command check_ssh
#}

define service{
```

```
    use     generic-service
    host_name       localhost
    service_description     ping 8.8.8.8
        check_command   check_google
}

define service{
    use     generic-service
    host_name       localhost
    service_description     DNS
        check_command   check_dns_local
}
```

Ficheiro de configuração 4: *(/etc/nagios3/conf.d/d1_nagios2.cfg)*

```
# A simple configuration file for monitoring the local host
# This can serve as an example for configuring other servers;
# Custom services specific to this host are added here, but services
# defined in nagios2-common_services.cfg may also apply.
#

define host{
    use             generic-host        ; Name of host template to use
    host_name           d1
    alias           d1
    address             192.168.137.11
    }

# Define a service to check the disk space of the root partition
# on the local machine.  Warning if < 20% free, critical if
# < 10% free space on partition.

define service{
    use                     generic-service     ; Name of service template to use
    host_name               d1
    service_description         Disk Space
    check_command               check_all_disks!20%!10%
    }

# Define a service to check the number of currently logged in
# users on the local machine.  Warning if > 20 users, critical
# if > 50 users.

define service{
    use                     generic-service     ; Name of service template to use
    host_name               d1
    service_description         Current Users
    check_command               check_users!20!50
    }

# Define a service to check the number of currently running procs
```

```
# on the local machine.  Warning if > 250 processes, critical if
# > 400 processes.

define service{
    use                     generic-service       ; Name of service template to use
    host_name               d1
    service_description         Total Processes
        check_command               check_procs!250!400
    }

# Define a service to check the load on the local machine.

define service{
    use                     generic-service       ; Name of service template to use
    host_name               d1
    service_description         Current Load
        check_command               check_load!5.0!4.0!3.0!10.0!6.0!4.0
    }
define service{
    use                     generic-service
    host_name               d1
    service_description         HTTP
        check_command               check_http
    }
define service{
    use                     generic-service
    host_name               d1
    service_description         SSH
        check_command        check_ssh
    }

define service{
    use    generic-service
    host_name     d1
    service_description MariaDB
        check_command   check_maria
}

define service{
    use    generic-service
    host_name     d1
    service_description    MariaDB Replicas
        check_command   check_replicas
}
```

Ficheiro de configuração 5: *(/etc/nagios3/conf.d/d2_nagios2.cfg)*

```
# A simple configuration file for monitoring the local host
# This can serve as an example for configuring other servers;
# Custom services specific to this host are added here, but services
# defined in nagios2-common_services.cfg may also apply.
```

```
#

define host{
    use             generic-host        ; Name of host template to use
    host_name       d2
    alias           d2
    address         192.168.137.12
    }

# Define a service to check the disk space of the root partition
# on the local machine.  Warning if < 20% free, critical if
# < 10% free space on partition.

define service{
    use                 generic-service     ; Name of service template to use
    host_name           d2
    service_description     Disk Space
    check_command           check_all_disks!20%!10%
    }



# Define a service to check the number of currently logged in
# users on the local machine.  Warning if > 20 users, critical
# if > 50 users.

define service{
    use                 generic-service     ; Name of service template to use
    host_name           d2
    service_description     Current Users
    check_command           check_users!20!50
    }

# Define a service to check the number of currently running procs
# on the local machine.  Warning if > 250 processes, critical if
# > 400 processes.

define service{
    use                 generic-service     ; Name of service template to use
    host_name           d2
    service_description     Total Processes
        check_command           check_procs!250!400
    }

# Define a service to check the load on the local machine.

define service{
    use                 generic-service     ; Name of service template to use
    host_name           d2
    service_description     Current Load
        check_command           check_load!5.0!4.0!3.0!10.0!6.0!4.0
    }
```

```
define service{
    use                     generic-service
    host_name               d2
    service_description     HTTP
        check_command               check_http
    }
define service{
    use                     generic-service
    host_name               d2
    service_description     SSH
        check_command       check_ssh
    }

define service{
    use    generic-service
    host_name     d2
    service_description MariaDB
        check_command   check_maria
}

define service{
    use    generic-service
    host_name     d2
    service_description    MariaDB Replicas
        check_command   check_replicas
}
```

Ficheiro de configuração 6: *(/etc/nagios3/conf.d/d3_nagios2.cfg)*

```
# A simple configuration file for monitoring the local host
# This can serve as an example for configuring other servers;
# Custom services specific to this host are added here, but services
# defined in nagios2-common_services.cfg may also apply.
#

define host{
    use             generic-host        ; Name of host template to use
    host_name           d3
    alias           d3
    address           192.168.137.13
    }

# Define a service to check the disk space of the root partition
# on the local machine.  Warning if < 20% free, critical if
# < 10% free space on partition.

define service{
    use                     generic-service      ; Name of service template to use
    host_name               d3
    service_description     Disk Space
    check_command             check_all_disks!20%!10%
```

```
    }

# Define a service to check the number of currently logged in
# users on the local machine.  Warning if > 20 users, critical
# if > 50 users.

define service{
    use                     generic-service       ; Name of service template to use
    host_name               d3
    service_description         Current Users
    check_command             check_users!20!50
    }

# Define a service to check the number of currently running procs
# on the local machine.  Warning if > 250 processes, critical if
# > 400 processes.

define service{
    use                     generic-service       ; Name of service template to use
    host_name               d3
    service_description         Total Processes
        check_command               check_procs!250!400
    }

# Define a service to check the load on the local machine.

define service{
    use                     generic-service       ; Name of service template to use
    host_name               d3
    service_description         Current Load
        check_command               check_load!5.0!4.0!3.0!10.0!6.0!4.0
    }
define service{
    use                     generic-service
    host_name               d3
    service_description         HTTP
        check_command               check_http
    }
define service{
    use                     generic-service
    host_name               d3
    service_description         SSH
        check_command         check_ssh
    }

define service{
    use    generic-service
    host_name     d3
    service_description MariaDB
        check_command   check_maria
}
```

```
define service{
      use     generic-service
      host_name      d3
      service_description     MariaDB Replicas
            check_command   check_replicas
}
```

DNS Server:

Neste projeto simula a distribuição de carga pelas duas gateways (bridge e bridge2)

cmd:

apt-get install bind9

Ficheiro de configuração: *(/etc/bind/db.sat.pt)*

```
;
; BIND reverse data file for local loopback interface
;
$TTL    604800
@       IN      SOA     sat.pt. root.sat.pt. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
      IN      NS      ns
ns    IN      A       192.168.137.1
www   IN      A       192.168.137.1;propria maquina
      IN      A       192.168.137.2;replica da maquina da bridge

/etc/bind/named.conf.default-zones
// prime the server with knowledge of the root servers
zone "." {
      type hint;
      file "/etc/bind/db.root";
};

// be authoritative for the localhost forward and reverse zones, and for
// broadcast zones as per RFC 1912

zone "localhost" {
      type master;
      file "/etc/bind/db.local";
};

zone "127.in-addr.arpa" {
      type master;
      file "/etc/bind/db.127";
};
```

```
zone "0.in-addr.arpa" {
	type master;
	file "/etc/bind/db.0";
};

zone "255.in-addr.arpa" {
	type master;
	file "/etc/bind/db.255";
};

zone "sat.pt" in{
	type master;
	file "/etc/bind/db.sat.pt";
};
```

Loganalyzer:

O LogAnalyser é um front-end para Syslog-Rsyslog de fácil compreensão e manuseio. É uma aplicação open source web-based, que foi desenvolvida maioritariamente em PHP. Permite uma análise muito prática dos vários logs do sistema através do web-browser.

No sistema SAT utiliza o arquivo texto gerado pelo Rsyslog.

(192.168.137.1:8080/loganalyzer/)

cmd:

 # apt-get install loganalyzer goaccess rkhunter

Ficheiro de configuração: *(/etc/rkhunter.conf)*

```
ROTATE_MIRRORS=1
UPDATE_MIRRORS=1
MIRRORS_MODE=0
MAIL-ON-WARNING=""
MAIL_CMD=mail -s "[rkhunter] Warnings found for ${HOST_NAME}"
TMPDIR=/var/lib/rkhunter/tmp
DBDIR=/var/lib/rkhunter/db
SCRIPTDIR=/usr/share/rkhunter/scripts
UPDATE_LANG=""
LOGFILE=/var/log/rkhunter.log
APPEND_LOG=0
COPY_LOG_ON_ERROR=0
COLOR_SET2=0
AUTO_X_DETECT=1
WHITELISTED_IS_WHITE=0
ALLOW_SSH_ROOT_USER=no
ALLOW_SSH_PROT_V1=0
ENABLE_TESTS="all"
DISABLE_TESTS="suspscan hidden_procs deleted_files packet_cap_apps apps"
USER_FILEPROP_FILES_DIRS="/var/www/*"
SCRIPTWHITELIST=/bin/egrep
SCRIPTWHITELIST=/bin/fgrep
```

SCRIPTWHITELIST=/bin/which
SCRIPTWHITELIST=/usr/bin/groups
SCRIPTWHITELIST=/usr/bin/ldd
SCRIPTWHITELIST=/usr/bin/lwp-request
SCRIPTWHITELIST=/usr/sbin/adduser
SCRIPTWHITELIST=/usr/sbin/prelink
IMMUTABLE_SET=0
PHALANX2_DIRTEST=0
ALLOW_SYSLOG_REMOTE_LOGGING=1
SUSPSCAN_TEMP=/dev/shm
SUSPSCAN_MAXSIZE=10240000
SUSPSCAN_THRESH=200
USE_LOCKING=0
LOCK_TIMEOUT=300
SHOW_LOCK_MSGS=1
DISABLE_UNHIDE=1
INSTALLDIR="/usr"

Rsyslog Server:

Ficheiro de configuração: (/etc/rsyslog.d/server.conf)

$ModLoad imuxsock
$ModLoad imklog
$ModLoad imudp
$UDPServerRun 514
$template FILENAME,"/var/log/%fromhost-ip%/syslog.log"
**.* ?FILENAME*

5. Bridge2

No mundo real esta máquina deveria ser igual à bridge no harware e no software instalado (exceto no ip). Estas máquinas (bridge e bridge2) funcionam em paralelo recebendo alternadamente os pedidos encaminhados pelo DNS server e encaminhando-os para as réplicas no caso de serem dinâmicos, ou devolvendo-os imediatamente ao cliente caso sejam estáticos.

Neste projeto utilizámos um portátil velhinho Dell (sem imagem no écran por não haver drivers compatíveis com a placa gráfica deste modelo) com NGINX e DNS só para provar que a solução funciona.

Com esta segunda máquina elimina-se um **ponto de falha grave**, que a acontecer iria parar todo o sistema.

Debian Server:
Download e instalação da versão mais recente em https://www.debian.org/distrib/

Rede Wireless:

Nesta máquina, o acesso à rede é feito através da placa de rede wireless interna do portátil, configurada em bridge na rede Eduroam da FCUL. Esta máquina faz assim a ponte entre as redes da Fcul (Eduroam) e do sistema SAT através da porta de rede Ethernet, ligada ao switch do sistema por cabo de rede.

cmd:

 # apt-get install wpa-supplicant

Ficheiro de configuração 1: *(/etc/init.d/eduroam.sh)*

```
case "$1" in
    start)
        wpa_supplicant -i wlan0 -c /etc/wpa_supplicant/eduroam.conf &
        dhclient wlan0
        ;;
    stop)
        echo "WTF?"
        ;;
    *)
        echo "OI?!"
        exit 1
        ;;
esac
exit 0
```

Ficheiro de configuração 2: *(/etc/wpa_supplicant/eduroam.conf)*

```
network={
    ssid="eduroam"
    key_mgmt=WPA-EAP
    eap=TTLS
    identity="fc41176@alunos.fc.ul.pt"
    password="---"
    phase1="peaplabel=1"
    phase2="auth=MSCHAPV2"
}
```

Interfaces de rede:

A configuração da interface de rede é necessária para a máquina poder comunicar com as outras máquinas do sistema SAT e ter acesso à internet para fazer download dos diferentes módulos necessários ao funcionamento do sistema, bem como updates do SO.

Ficheiro de configuração: (*/etc/network/interfaces*)

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).
```

 # The loopback network interface
auto lo
iface lo inet loopback

 # The primary network interface
auto eth0
iface eth0 inet static
address 192.168.137.2
netmask 255.255.255.0
gateway 10.101.97.1

Atualizar servidor:
Instalação dos updates do Debian. Essencial para a performance e segurança do sistema.

cmd:
 # sudo apt-get update
 # sudo apt-get upgrade

Hosts:
O ficheiro hosts faz o relacionamento entre um nome de computador e um endereço IP. Torna a operação e configuração do sistema mais simplificada e rápida.

Ficheiro de configuração: (/etc/hosts)

127.0.0.1 localhost
127.0.1.1 bridge2
192.168.137.11 d1
192.168.137.12 d2
192.168.137.13 d3

The following lines are desirable for IPv6 capable hosts
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

NGINX Server + Sticky Module:
Nginx é um webserver open source, um reverse proxy server para protocolos HTTP, SMTP, POP3 e IMAP e um application load balancer com um forte foco na performance e no baixo uso de memória e com inúmeras possibilidades de configuração para melhor performance.

Tecnicamente, o Nginx consome menos memória que o Apache, pois lida com requisições Web através do conceito de “event-based web server” (asynchronous) , já o Apache é baseado no conceito “process-based server” utilizando threads para tratar dos pedidos.

O servidor Web Nginx garante o balanceamento de carga pelas 3 réplicas (utilizando o algoritmo Round Robin) nos pedidos para conteúdos dinâmicos (pedidos à bases de dados) e responde imediatamente aos pedidos de conteúdos estáticos.

O módulo adicional Sticky, garante que os pedidos dinâmicos de uma sessão sejam servidos sempre a partir da mesma réplica.

DOWNLOAD DO NGINX DO SITE OFFICIAL, DOWNLOAD STICKY MODULE (https://github.com/lusis/nginx-sticky-module)

cmd:
 # apt-get install libssl-dev
 # apt-get install zlib1g-dev
 # apt-get install libpcre3-dev
 # apt-get source nginx
./configure --without-mail_smtp_module --without-mail_pop3_module --without-mail_imap_module --add-module=/home/user/nginx-sticky-module-1.1

Ficheiro de configuração: (/etc/nginx/sites-enabled/default)

```
upstream frontend{
    server d1:80;
    server d2:80;
    server d3:80;
    sticky;
}
server{
    location /static/ {
        root /var/www/;
    }
    location / {
        proxy_pass http://frontend;
    }
    location /RequestDenied {
        return 418;
    }
}
```

DNS Server:
Neste projeto simula a distribuição de carga pelas duas gateways (bridge e bridge2)

cmd:
 # apt-get install bind9

Ficheiro de configuração: *(/etc/bind/db.sat.pt)*

```
;
; BIND reverse data file for local loopback interface
;
$TTL    604800
@       IN      SOA     sat.pt. root.sat.pt. (
```

```
                    1       ; Serial
                604800      ; Refresh
                 86400      ; Retry
                2419200     ; Expire
                604800 )    ; Negative Cache TTL
;
    IN    NS    ns
ns    IN    A     192.168.137.1
www   IN    A     192.168.137.1;propria máquina
    IN    A     192.168.137.2;replica da maquina da bridge

/etc/bind/named.conf.default-zones
// prime the server with knowledge of the root servers
zone "." {
    type hint;
    file "/etc/bind/db.root";
};

// be authoritative for the localhost forward and reverse zones, and for
// broadcast zones as per RFC 1912

zone "localhost" {
    type master;
    file "/etc/bind/db.local";
};

zone "127.in-addr.arpa" {
    type master;
    file "/etc/bind/db.127";
};

zone "0.in-addr.arpa" {
    type master;
    file "/etc/bind/db.0";
};

zone "255.in-addr.arpa" {
    type master;
    file "/etc/bind/db.255";
};

zone "sat.pt" in{
    type master;
    file "/etc/bind/db.sat.pt";
};
```