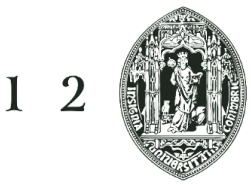


UNIVERSIDADE D
COIMBRA

Trabalho 3

Monitorização e alta disponibilidade

Duarte Manuel Bento Dias 2018293526
André Pinto Dias 2020209083

Índice

Índice **2**

Introdução e objetivo **2**

Ferramentas e Software **3**

Arquitetura de Rede **4**

DNS **5**

Apache **6**

Alta Disponibilidade **7**
- **VRRP** **7**
- **HAProxy** **8**

Zabbix **9**
- **Zabbix Mainframe** **10**
  - Configuração de clientes e templates 10
    - Notas de instalação 11
      - (1) Apache 11
      - (2) HAProxy 11
  - Ecrãs 12
  - Mapa de rede 13
  - Alertas 13

Monitorização Web **16**
- Autenticação 17
- Funcionamento 17

Testes e Funcionamento **18**
- Falha de servers web e análise do tempo de recuperação 18
- Falhas na HAProxy e tempo de recuperação 18
- Falhas e respetiva identificação no Zabbix 19
- Impacto de um número elevado de pedidos no juice shop 19
- Alertas no Zabbix 20
- Correlacionar CPU e tráfego com o zabbix 20
  - Utilização do CPU 20
  - Utilização de rede 20

Introdução e objetivo

Este trabalho foi realizado no âmbito da cadeira GISI com intuito de configurar uma infraestrutura de rede de alta disponibilidade, bem como um serviço de monitorização de rede. Neste projeto implementa-se uma arquitetura simples na qual existem 2 servidores que fazem de frontend, numa espécie de reverse-proxy, e 2 servidores web usados em round robin.

Ferramentas e Software

Neste trabalho foram utilizadas as seguintes packages:

- Docker;
- Zabbix;
- VRRP (Keepalived);
- HAProxy;
- Apache2;
- OWASP Juice Shop.

Arquitetura de Rede

Para a realização deste trabalho foi configurada a topologia de rede demonstrada na Figura 1.

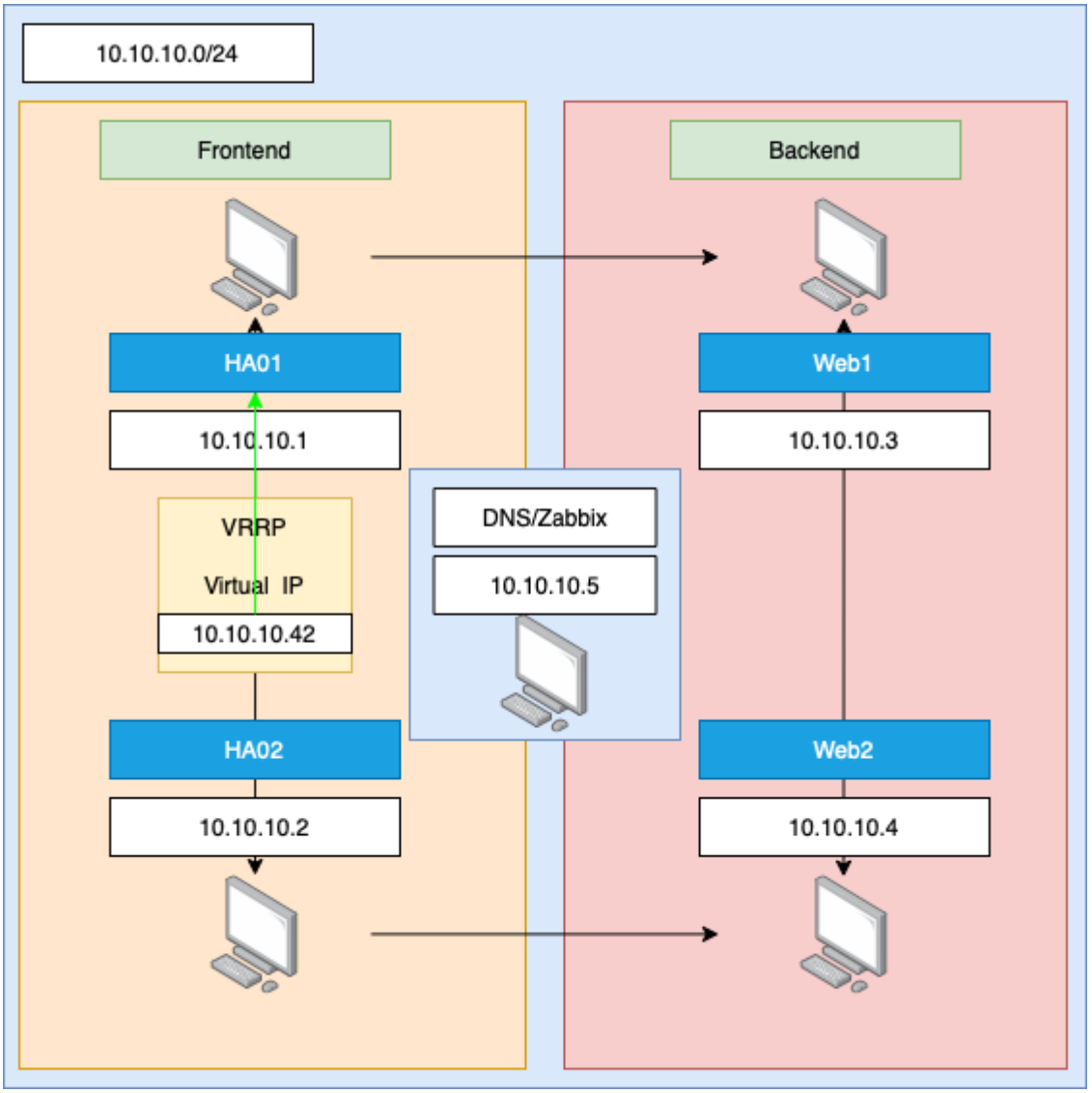


Fig. 1 Arquitetura de Rede

DNS

A configuração de DNS segue um esquema semelhante aos trabalhos anteriores, sendo utilizado o serviço bind.

```
#File /etc/bind/gisi.db
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     dns.gisi.pt. webmaster.gisi.pt. (
                         2022030601     ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                         2419200        ; Expire
                          604800 )      ; Negative Cache TTL
;
                IN      NS      dns

@               IN      A       10.10.10.42
ha-01           IN      A       10.10.10.1
ha-02           IN      A       10.10.10.2
web1            IN      A       10.10.10.3
web2            IN      A       10.10.10.4
dns             IN      A       10.10.10.5
```

```
#File /etc/bind/gisi.db
; BIND reverse data file for local loopback interface
;
$TTL    604800
@       IN      SOA     dns.gisi.pt. webmaster.gisi.pt. (
                     2022030900         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                         2419200        ; Expire
                          604800 )      ; Negative Cache TTL
;
@       IN      NS      dns.gisi.pt.

dns.gisi.pt.    IN      A       10.10.10.5

1       IN      PTR     ha-01.gisi.pt.
2       IN      PTR     ha-02.gisi.pt.
```

```
3       IN      PTR     web1.gisi.pt.
4       IN      PTR     web2.gisi.pt.
5       IN      PTR     dns.gisi.pt.
42       IN      PTR     gisi.pt.
```

```
tldart@HA1:~$ ping gisi.pt
PING gisi.pt (10.10.10.42) 56(84) bytes of data.
64 bytes from 10.10.10.42: icmp_seq=1 ttl=64 time=0.107 ms
64 bytes from 10.10.10.42: icmp_seq=2 ttl=64 time=0.139 ms
64 bytes from 10.10.10.42: icmp_seq=3 ttl=64 time=0.177 ms
64 bytes from 10.10.10.42: icmp_seq=4 ttl=64 time=0.171 ms
64 bytes from 10.10.10.42: icmp_seq=5 ttl=64 time=0.059 ms
64 bytes from 10.10.10.42: icmp_seq=6 ttl=64 time=0.176 ms
64 bytes from 10.10.10.42: icmp_seq=7 ttl=64 time=0.178 ms
```

Fig. 2 Funcionamento do DNS

Apache

O primeiro passou pela configuração dos servidores apache com recurso à package apache2. Fez-se apenas uma ligeira mudança no conteúdo do website (Fig 3).

Numa situação realista, ambos os servidores apresentam o mesmo conteúdo web, combinando tanto o VRRP como o HAProxy (mencionado abaixo) para garantir que, no caso de algum tipo de falha num dos aparelhos, o cliente final não é afetado.

Neste caso, e de forma a validar o funcionamento do HAProxy, as páginas web foram configuradas de forma a serem distinguidas.

```
#File /var/www/html/index.html
Never Gonna Give You Up
```

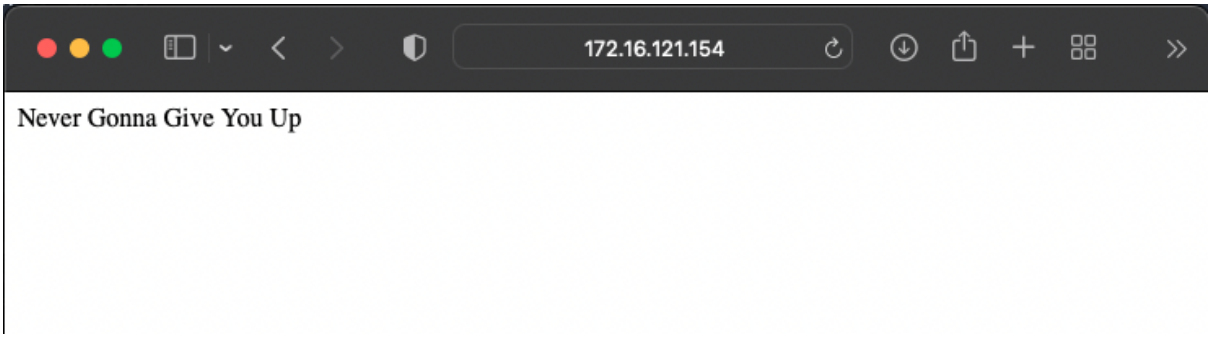


Fig. 3 Web1

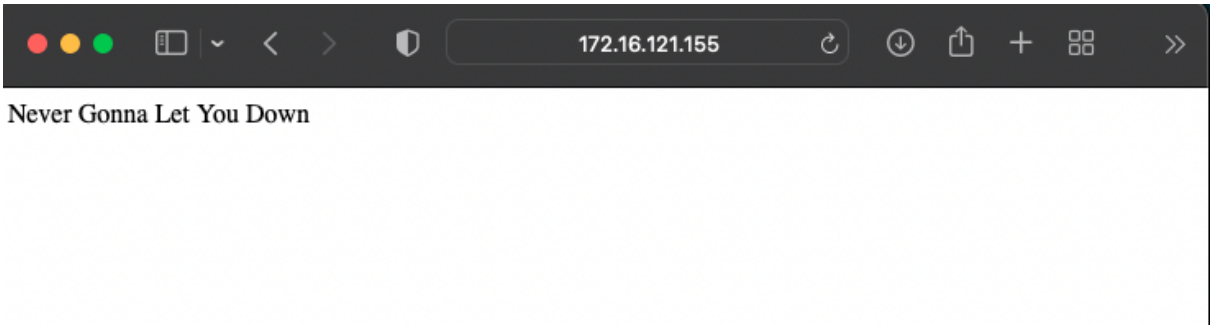


Fig. 4 Web2

Alta Disponibilidade

VRRP

O *Virtual Router Redundancy Protocol* é um protocolo de rede que permite associação automática de IP 's a servidores. Este serviço é bastante útil no caso de falhas de rede, dado que utilizando o VRRP é possível trocar o servidor recipiente dos dados sem que o cliente se aperceba.

Neste trabalho foi configurado VRRP entre o host HA-01 e o host HA-02, utilizando a package *keepalived*.

```
#MASTER
#File /etc/keepalived/keepalived.conf
vrrp_instance VRRP_1{
        state MASTER
        interface ens256
        virtual_router_id 42
        priority 200
        advert_int 1
        authentication{
                auth_type PASS
                auth_pass 1066
        }
        virtual_ipaddress{
                10.10.10.42/24
        }
}
```

```
#BACKUP
#/etc/keepalived/keepalived.conf
```

```
vrrp_instance VRRP_1{
        state BACKUP
        interface ens256
        virtual_router_id 42
        priority 100
        advert_int 1
        authentication{
                auth_type PASS
                auth_pass 1066
        }
        virtual_ipaddress{
                10.10.10.42/24
        }
}
```

No servidor ativo (ou, no caso de ambos estarem ativos, no master) é possível verificar que, para além do IP atribuído fisicamente à placa de rede, temos também um IP virtual, endereço este atribuído pelo protocolo VRRP.

```
3: ens256: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:be:e7:11 brd ff:ff:ff:ff:ff:ff
    altname enp26s0
    inet 10.10.10.1/24 brd 10.10.10.255 scope global ens256
       valid_lft forever preferred_lft forever
    inet 10.10.10.42/24 scope global secondary ens256
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:febe:e711/64 scope link
       valid_lft forever preferred_lft forever
```

Fig. 3 Funcionamento do HAProxy

HAProxy

O HAProxy é outra ferramenta que visa garantir máxima disponibilidade. Este serviço funciona como um reverse proxy, permitindo assim uma distribuição de carga pelos servidores web disponíveis, neste caso o Web1 e o Web2.

Neste caso, em ambos os servidores proxy foi adicionada a seguinte configuração.

```
#File /etc/haproxy/haproxy.cfg
frontend apache_front
        # Frontend listen port - 80
        bind *:80
        # Set the default backend
        default_backend     apache_backend_servers
        # Enable send X-Forwarded-For header
        option              forwardfor
```

```
# Define backend
backend apache_backend_servers
        # Use roundrobin to balance traffic
        balance             roundrobin
        # Define the backend servers
        server              backend01 10.10.10.3:80 check
        server              backend02 10.10.10.4:80 check
```

Na Fig. 4 conseguimos observar o funcionamento do HAProxy, em Round Robin, no acesso ao site.

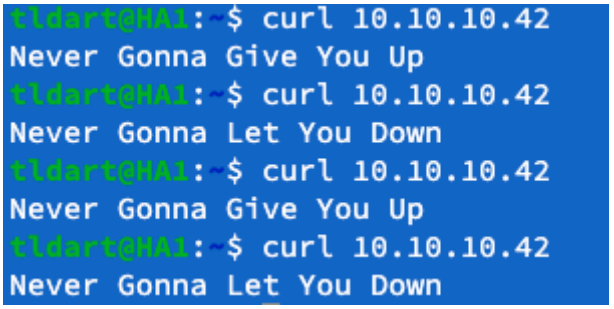


Fig. 4 Funcionamento do HAPRoxy

Zabbix

O Zabbix é um serviço de monitorização da rede e dos seus respetivos dispositivos. Este serviço assenta numa arquitetura cliente-servidor, onde existe um manager e vários zabbix-agent (clientes). O zabbix manager integra, normalmente, uma interface web.

No nosso caso o zabbix manager foi instalado num docker no servidor de DNS, e pode ser consultado na porta 8080. Os clientes Zabbix são instalados diretamente na máquina destino utilizando a package *zabbix-agent.*

A instalação dos serviços é trivial, sendo seguidos os passos da PL7. O ficheiro de configuração incluído em todos os clientes monitorizados é apresentado abaixo.

```
#File /etc/zabbix/zabbix_agentd.conf
PidFile=/run/zabbix/zabbix_agentd.pid
LogFile=/var/log/zabbix-agent/zabbix_agentd.log

LogFileSize=0
Server=0.0.0.0/0

ListenPort=10050

ServerActive=10.10.10.5
```

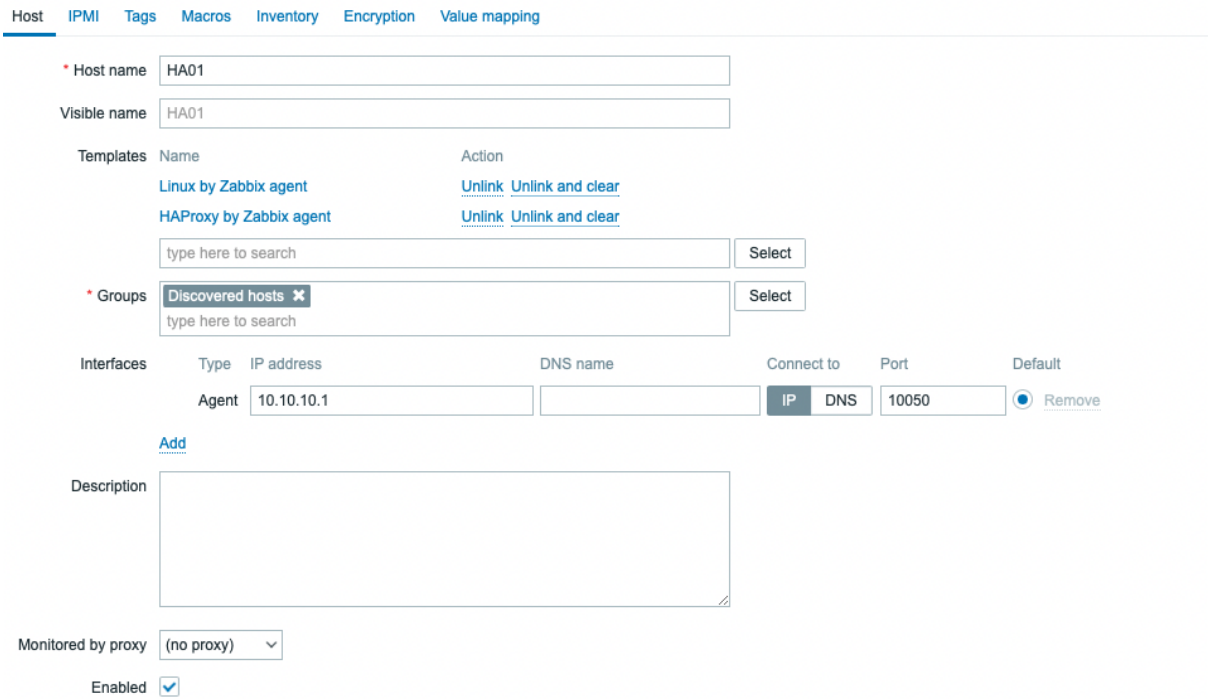
```
Hostname=HA01

Include=/etc/zabbix/zabbix_agentd.conf.d/*.conf
```

Zabbix Mainframe

Configuração de clientes e templates

Depois de configurados os respectivos zabbix-agent, estes são adicionados no frontend do zabbix.



Como pedido, foram selecionados alguns templates aplicáveis, nomeadamente:

- Linux by Zabbix Agent:
    - Aplicados a todos os sistemas que envergam plataforma linux;
- Apache by HTTP:
    - Que monitoriza package apache instalada anteriormente 1);
- HAproxy by Zabbix:
    - Monitoriza o serviço HAProxy(2).

Notas de instalação

(1) Apache

Como referido em https://www.zabbix.com/integrations/apache é necessário adicionar:

```
#File  /etc/apache2/apache2.conf
<Location "/server-status">
SetHandler server-status
Require host gisi
</Location>
```

É necessário também efetuar algumas alterações de forma a permitir conexões ao à página /server-status a partir da mesma rede(https://techexpert.tips/apache/apache-mod_status-installation/).

```
#File /etc/apache2/mods-enabled/status.conf
<Location /server-status>
                SetHandler server-status
                Require local
                Require ip 10.10.10.0/24
 </Location>
```

```
#File /etc/apache2/sites-enabled/000-default.conf
<Location /server-status>
        SetHandler server-status
        Require local
        Require ip 10.10.10.0/24
</Location>
```

(2) HAProxy

No caso do HAProxy, segundo https://www.zabbix.com/integrations/haproxy, é necessário adicionar:

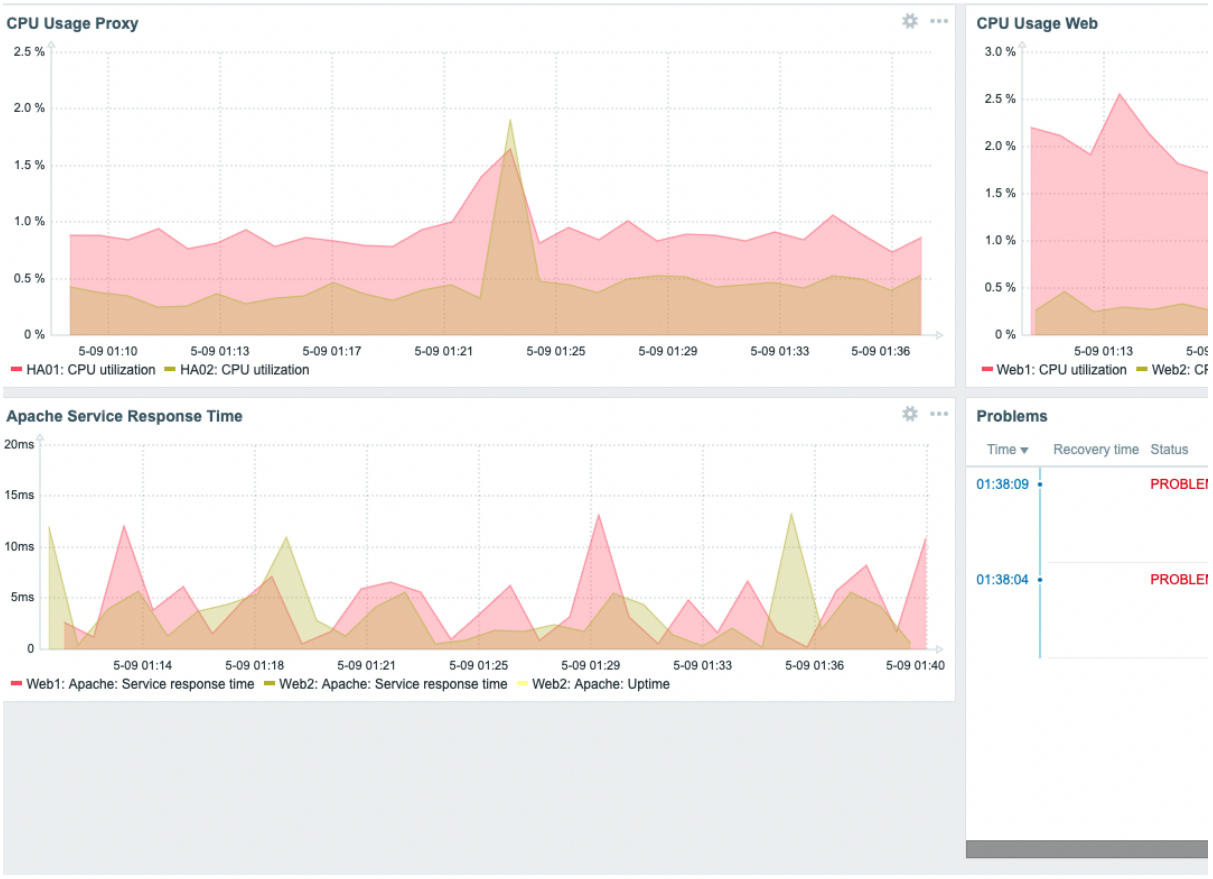
```
#File /etc/haproxy/haproxy.conf
frontend stats
    bind *:8404
    stats enable
    stats uri /stats
    stats refresh 10s
```

Após a configuração dos agentes, o panorama geral é:

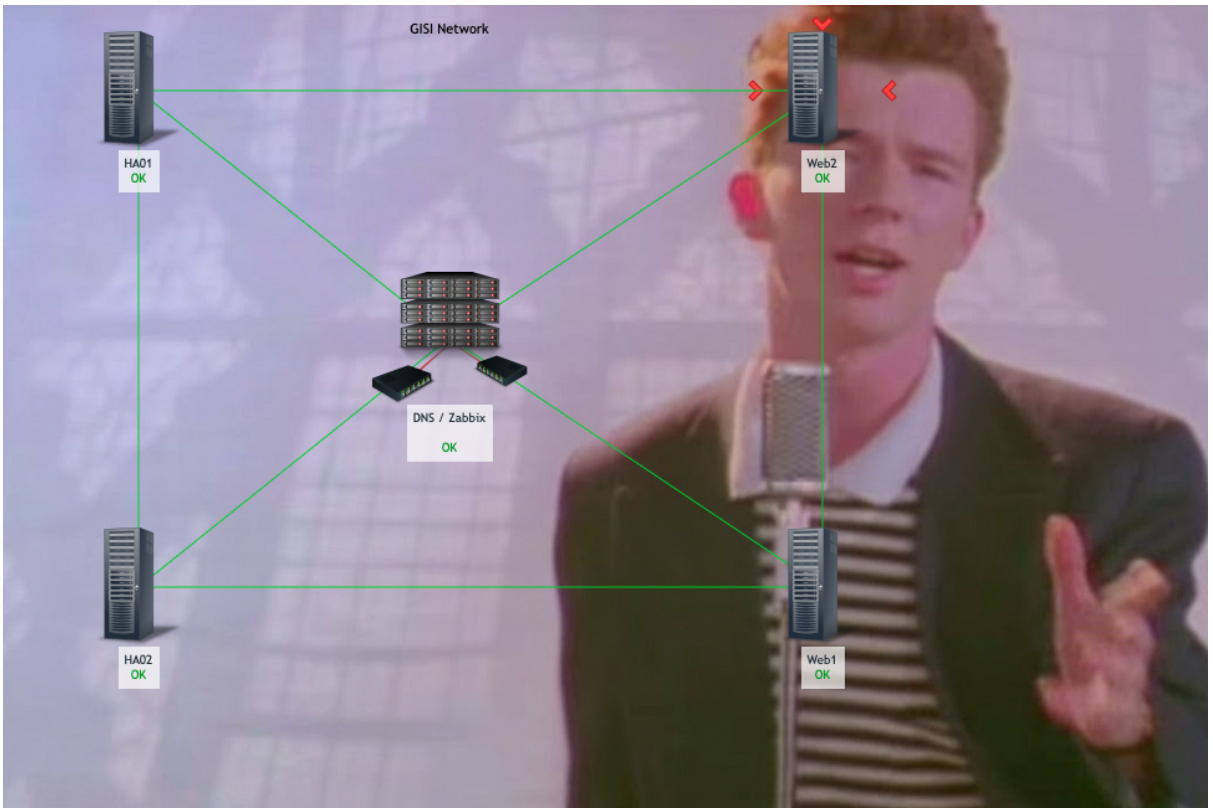
Name ▲	Items	Triggers	Graphs	Discovery	Web	Interface	Proxy	Templates
DNS	Items 100	Triggers 30	Graphs 23	Discovery 3	Web	10.10.10.5:10050		Linux by Zabbix agent
HA01	Items 107	Triggers 37	Graphs 23	Discovery 6	Web	10.10.10.1:10050		HAProxy by Zabbix agent, Linux by Zabbix agent
HA02	Items 107	Triggers 34	Graphs 23	Discovery 6	Web	10.10.10.2:10050		HAProxy by Zabbix agent, Linux by Zabbix agent
Web1	Items 122	Triggers 35	Graphs 26	Discovery 4	Web	10.10.10.3:10050		Apache by HTTP, Linux by Zabbix agent
Web2	Items 122	Triggers 35	Graphs 26	Discovery 4	Web	10.10.10.4:10050		Apache by HTTP, Linux by Zabbix agent

Ecrãs

A criação de menus de monitorização é feita através da interface gráfica web.
Monitoring > Dashboard > New Dashboard



Mapa de rede



Alertas

O esquema de notificações utilizou-se a plataforma Telegram. Foi seguido o tutorial https://git.zabbix.com/projects/ZBX/repos/zabbix/browse/templates/media/discord.
Os passos essenciais são:

- Criação do Bot na plataforma telegram;
- Criação de um trigger no zabbix (não necessário, pode ser utilizado um trigger pré-construído.
- Configuração do API do Telegram;
- Configuração de um user e adição de um media type;
- Configuração de uma action;

Administration > Media_types > Telegram:

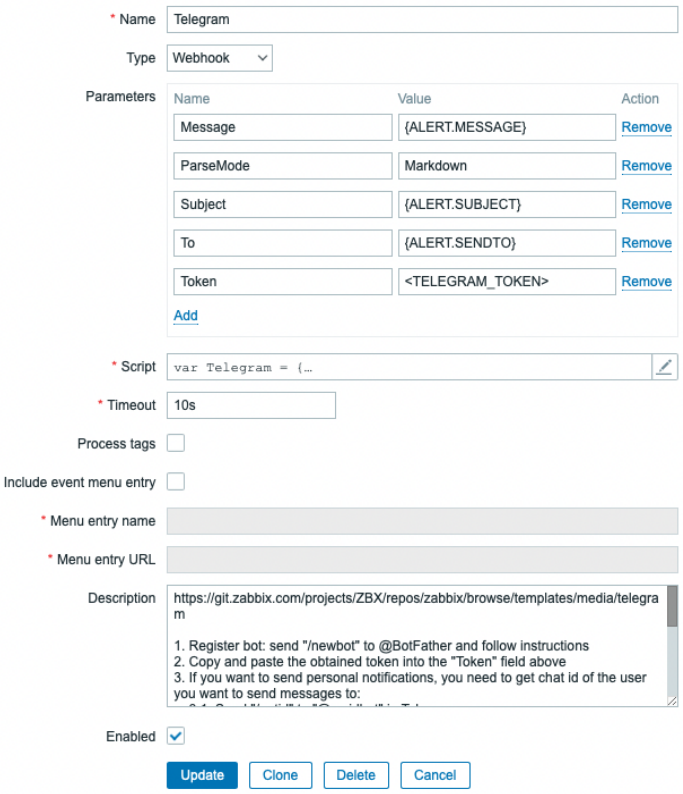


Fig. 5 Menu Zabbix Telegram

Administration > Users > Create User > Media:

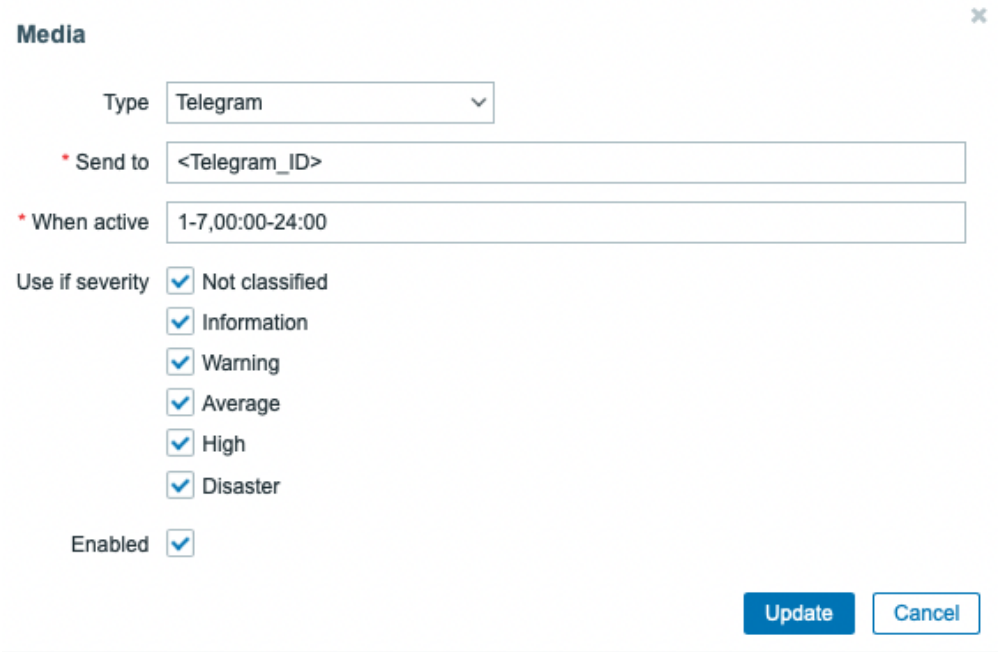


Fig. 6 Menu Zabbix User

Configuration > Actions > Trigger Actions > Create Action:

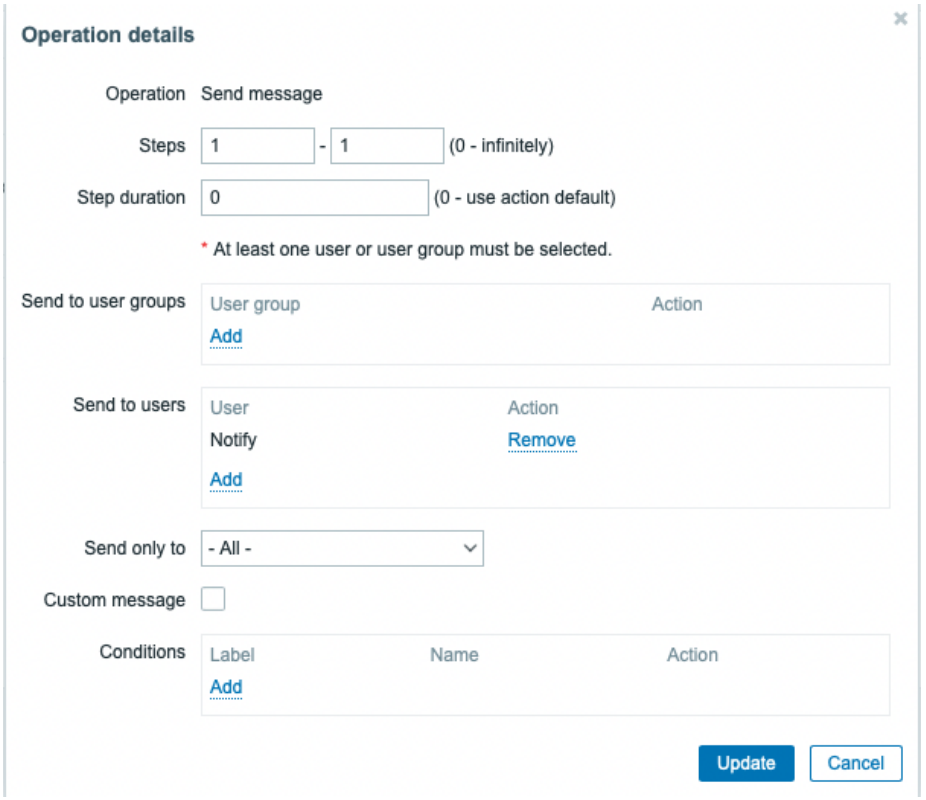
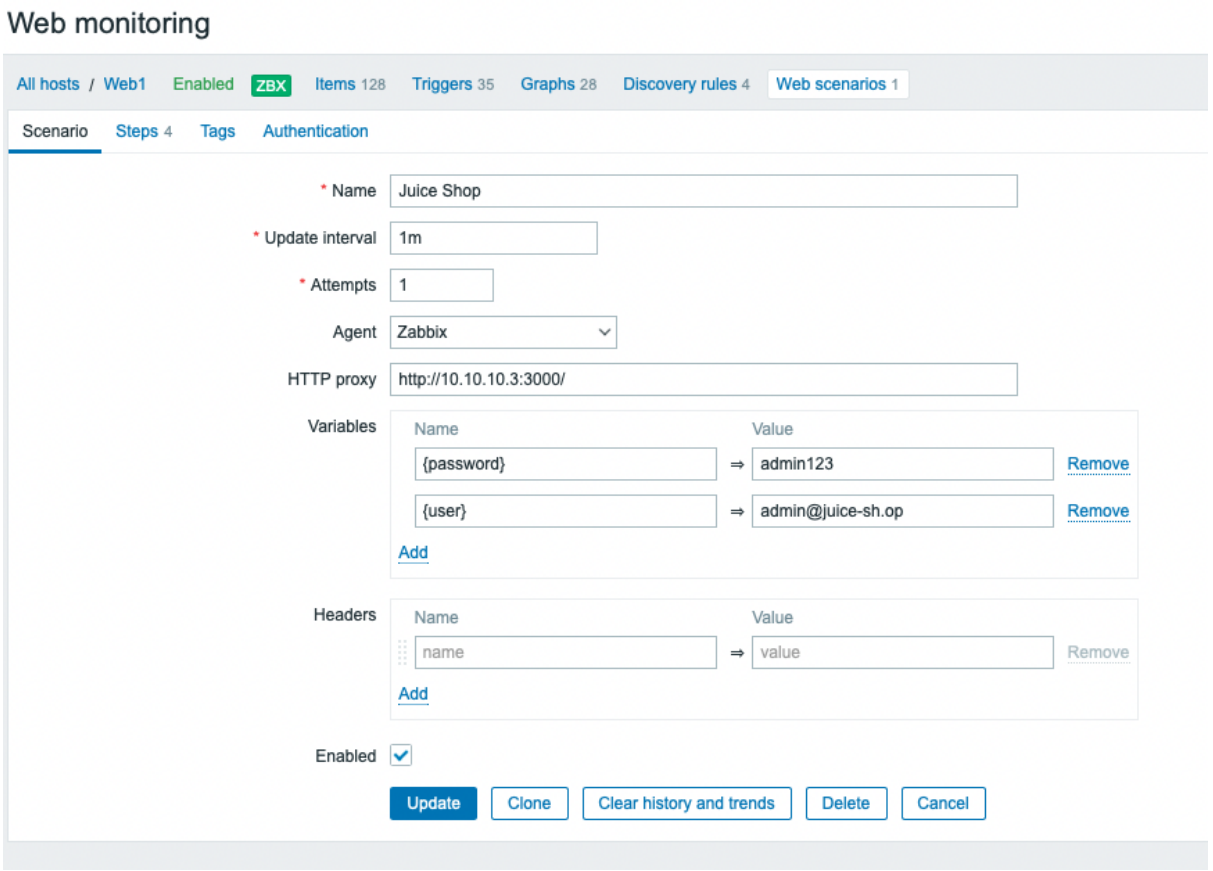


Fig. 7 Menu Zabbix Triggers

(Nota: Não esquecer alterar o estado da actions para enabled)

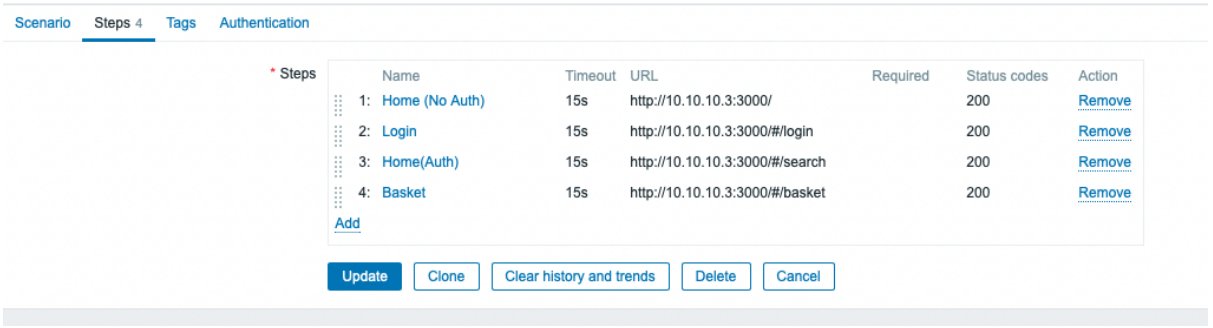
Monitorização Web

A monitorização Web fez uso da aplicação JuiceShop. Esta foi instalada no host Web1 através da plataforma docker, utilizando a porta 3000.



Para fazer a monitorização em detalhe do site foram configurados os seguintes passos:

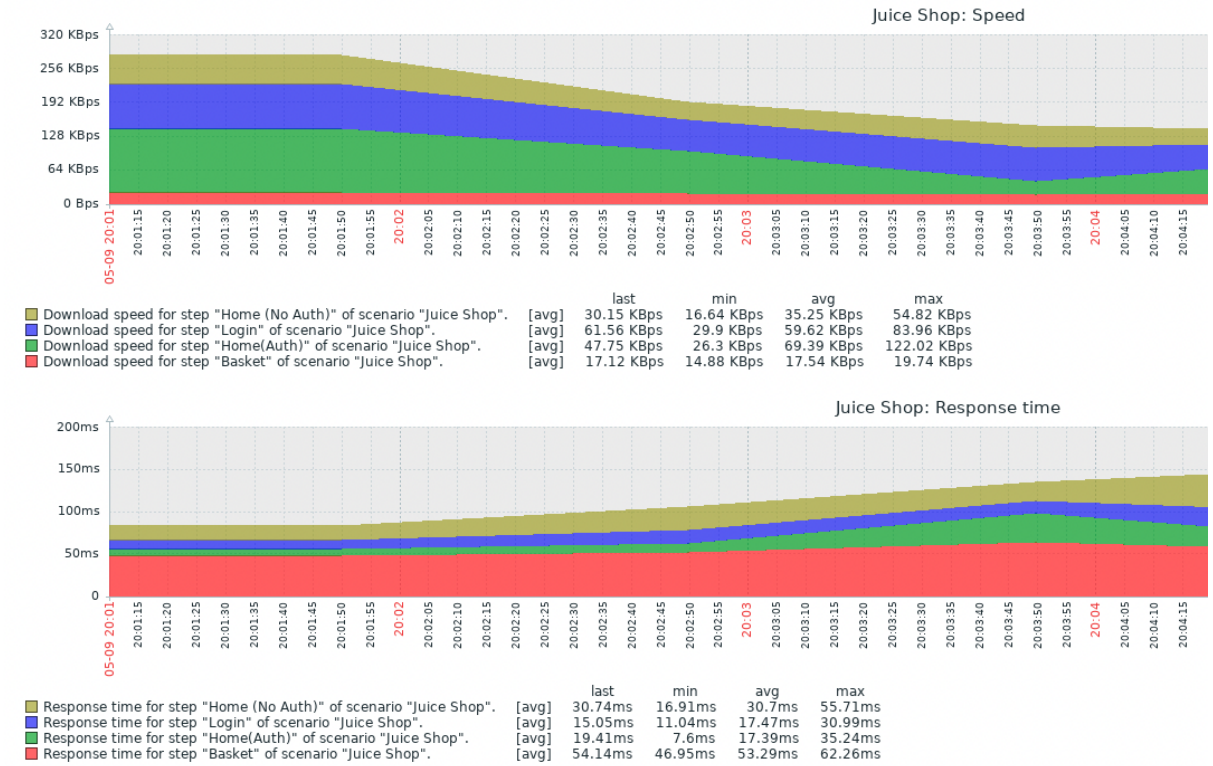
- Home (Non Authenticated)– Monitoriza a *home page* sem autenticação
- Log in – Monitoriza a página de login.
- Home (Authenticated) - Monitoriza também a pagina inicial mas agora com autenticação
- Product Details (Authenticated) - Monitoriza a lista de produtos



Autenticação

Nas páginas em que é necessário autenticação faz-se a recolha do token, sendo este posteriormente utilizado no campo header.

Funcionamento



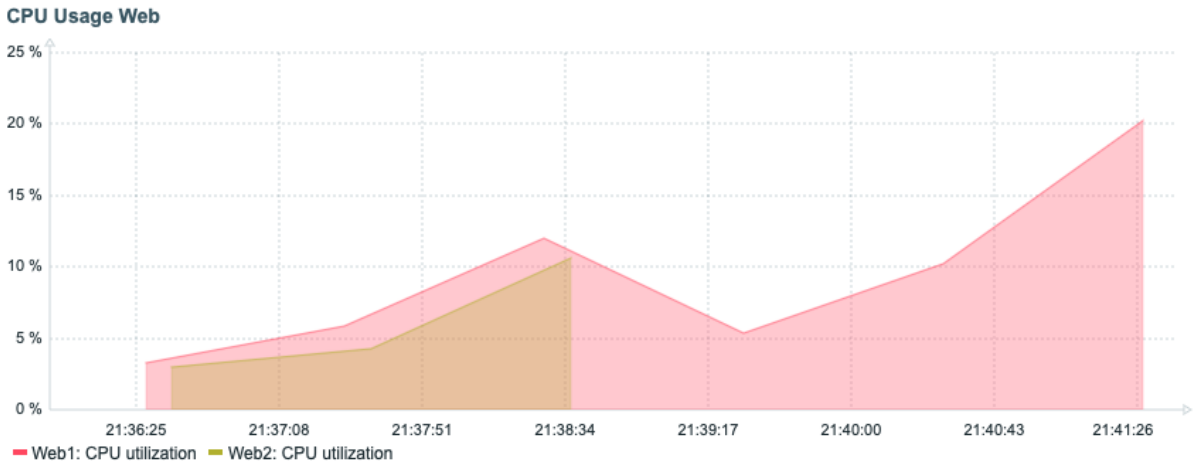
Testes e Funcionamento

Falha de servers web e análise do tempo de recuperação

Condição: Desligar Web2

```
ab -k -c 1 -n 200000 http://10.10.10.42/
```

Inicialmente temos ambos os servidores ligados, depois apenas o Web1.



Falhas na HAProxy e tempo de recuperação

Condição: Desligar HA01 (Nota-se apenas um pequeno aumento numa das respostas icmp, 1.37ms)

```
PING 10.10.10.42 (10.10.10.42) 56(84) bytes of data.
64 bytes from 10.10.10.42: icmp_seq=1 ttl=64 time=0.819 ms
64 bytes from 10.10.10.42: icmp_seq=2 ttl=64 time=0.369 ms
64 bytes from 10.10.10.42: icmp_seq=3 ttl=64 time=0.331 ms
64 bytes from 10.10.10.42: icmp_seq=4 ttl=64 time=0.442 ms
64 bytes from 10.10.10.42: icmp_seq=5 ttl=64 time=0.397 ms
64 bytes from 10.10.10.42: icmp_seq=6 ttl=64 time=0.604 ms
64 bytes from 10.10.10.42: icmp_seq=10 ttl=64 time=1.37 ms
64 bytes from 10.10.10.42: icmp_seq=11 ttl=64 time=0.757 ms
64 bytes from 10.10.10.42: icmp_seq=12 ttl=64 time=0.324 ms
```

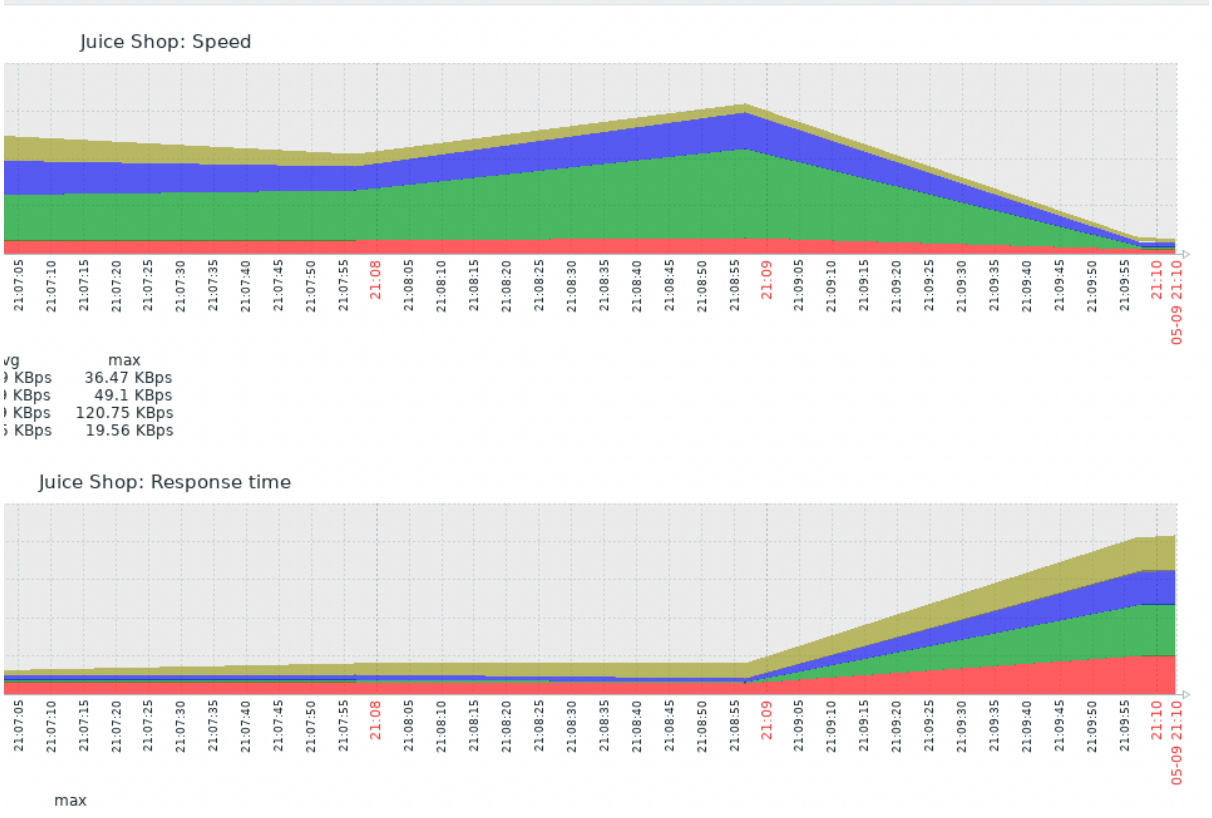
Falhas e respetiva identificação no Zabbix

Problems

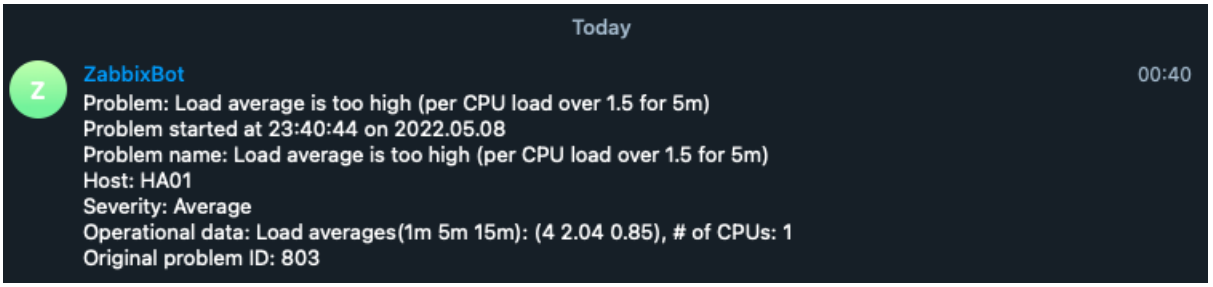
Time ▾	Recovery time	Status	Info	Host	Problem • Severity	Duration	Ack
21:14:11		PROBLEM		HA02	Zabbix agent is not available (for 3m)	18s	No
21:13:07		PROBLEM		Web2	Apache: Service is down	1m 22s	No

Impacto de um número elevado de pedidos no juice shop

```
ab -k -c 350 -n 200000 http://10.10.10.3:3000/
```



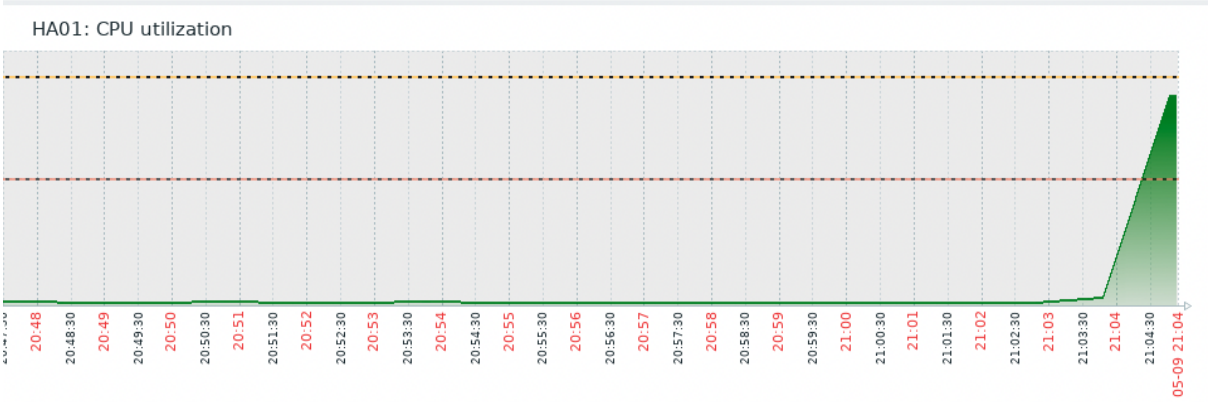
Alertas no Zabbix



Correlacionar CPU e tráfego com o zabbix

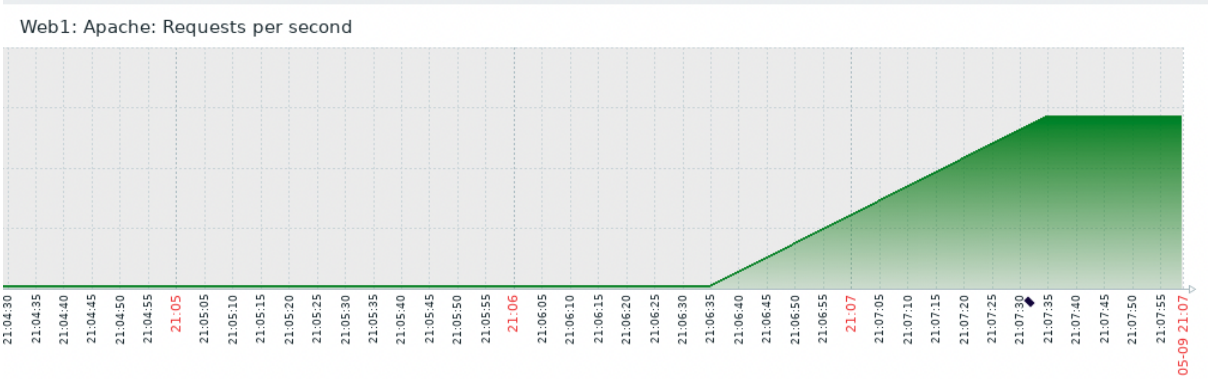
Utilização do CPU

```
yes >> /dev/null &
```



Utilização de rede

```
ab -k -c 350 -n 20000 http://10.10.10.3/
```



Referências:

https://adminscriptbank.wordpress.com/2016/09/16/debian-install-and-configure-vrrp-with-keepalived/
https://httpd.apache.org/docs/2.4/programs/ab.html
https://www.zabbix.com/documentation/current/en/manual/web_monitoring/example
https://www.zabbix.com/documentation/current/en/manual/web_monitoring
https://github.com/zabbix/community-templates/tree/main/Applications/DNS/template_bind_stat
https://www.zabbix.com/integrations/haproxy#haproxy_agent
https://git.zabbix.com/projects/ZBX/repos/zabbix/browse/templates/media/discord