

# Gestão de Redes com Nagios e Zabbix



Mestrado Integrado em Engenharia Electrotécnica e  
Computadores

Planeamento e Gestão de Redes

Francisco Fernandes Xavier de Barros – 201506338  
João Nuno Barbosa Neves – 201405198

30 de Maio de 2019

# Introdução

No âmbito da unidade curricular de Planeamento e Gestão de Redes foram estudadas para este trabalho duas ferramentas de gestão de equipamentos e serviços de uma rede, e as suas componentes de monitorização, sendo elas o Nagios e o Zabbix. Para a monitorização foram configurados, um servidor Web, um servidor FTP, um servidor NTP, um servidor de e-mail e um servidor cache de DNS.

Para obter resultados de bom funcionamento foram forçados erros nos servidores tal como pedido no guião.

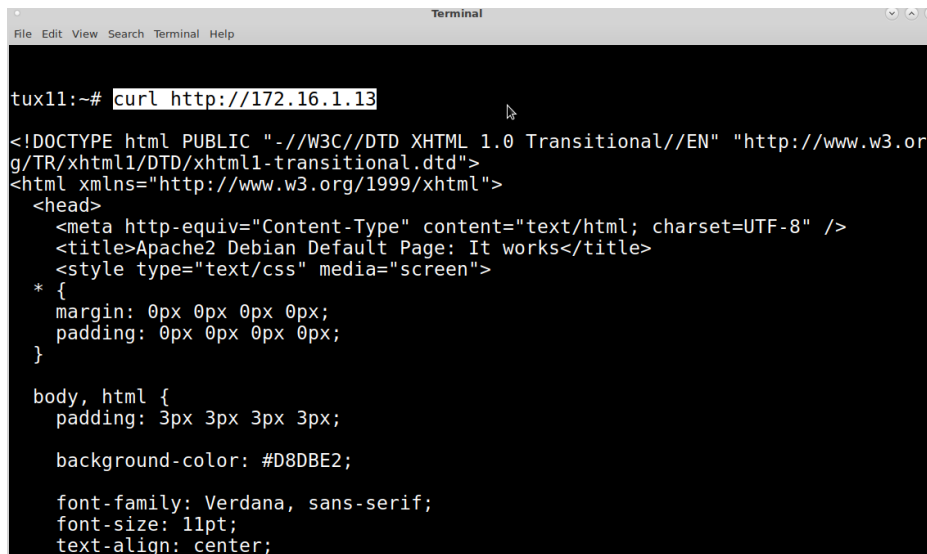
## Configuração dos Servidores

Optámos por distribuir os servidores pelos tux's disponíveis da seguinte forma:

- tux11 - Servidor de mail postfix
- tux12 - Servidor de cache DNS
- tux13 - Servidores de NTP, FTP e Apache
- tux14 - Nagios e Zabbix

### Servidor Web

Para o servidor web fizemos o download do pacote apache2 que se configura automaticamente de uma forma suficiente para o efeito. Observe-se na imagem abaixo o pedido feito ao mesmo a partir do tux11:



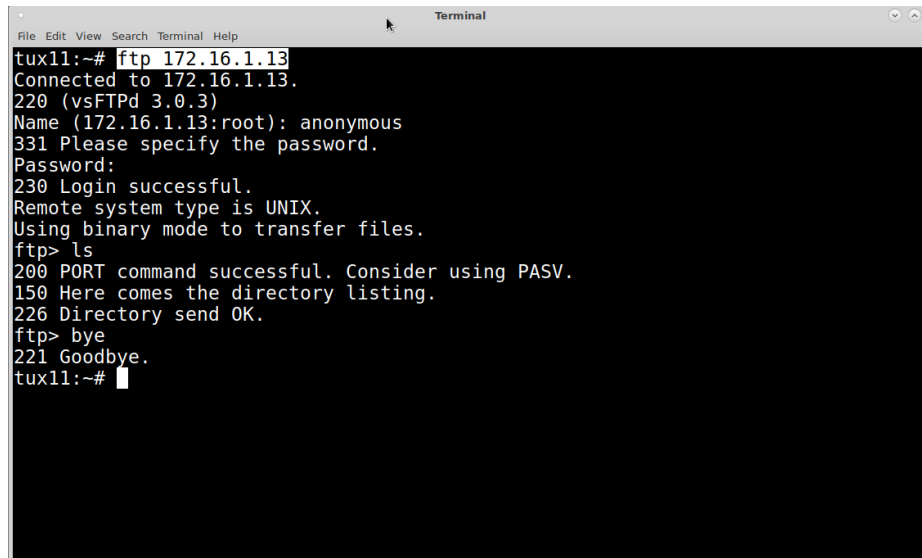
```
tux11:~# curl http://172.16.1.13
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Apache2 Debian Default Page: It works</title>
    <style type="text/css" media="screen">
      * {
        margin: 0px 0px 0px 0px;
        padding: 0px 0px 0px 0px;
      }
      body, html {
        padding: 3px 3px 3px 3px;
        background-color: #D8DBE2;
        font-family: Verdana, sans-serif;
        font-size: 11pt;
        text-align: center;
      }
    </style>
  </head>
  <body>
    <div style="text-align: center;">
      <img alt="Apache2 logo" data-bbox="488 665 511 688" style="height: 1em; margin: 0 auto 0 auto;"/>
    </div>
  </body>
</html>
```

## Servidor FTP

Para o servidor FTP foram executados os seguintes comandos:

- apt install vsftpd
- nano /etc/vsftpd.conf
  - Configurações acrescentadas:
    - anonymous\_enable = YES
    - anon\_upload\_enable = YES
    - write\_enable = YES
    - anon\_mkdir\_write\_enable = YES
- systemctl restart vsftpd
- systemctl enable vsftpd

Para obtermos tráfego FTP criámos um pequeno script em python (ftp\_python.py, em anexo) que faz um login anónimo e uma listagem do diretório. Observe-se embaixo um acesso a partir do tux11:



```
tux11:~# ftp 172.16.1.13
Connected to 172.16.1.13.
220 (vsFTPd 3.0.3)
Name (172.16.1.13:root): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
226 Directory send OK.
ftp> bye
221 Goodbye.
tux11:~#
```

## Servidor NTP

Para este servidor instalámos o pacote NTP no tux13 e o pacote ntpdate nos restantes tux's. Com o comando que se demonstra embaixo vemos o bom funcionamento do servidor:

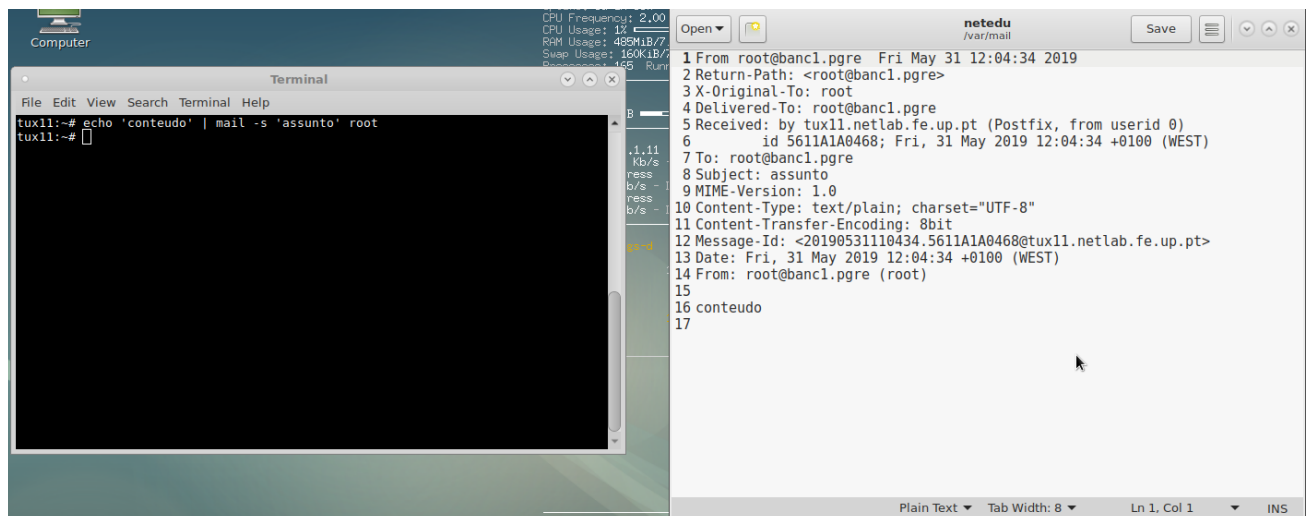
```
Terminal
File Edit View Search Terminal Help
tux11:~# ntpdate -u 172.16.1.13
31 May 12:06:51 ntpdate[10400]: adjust time server 172.16.1.13 offset 0.001553 s
ec
tux11:~#
```

## Servidor e-mail

O servidor de e-mail necessitou de um número maior de comandos, sendo eles:

- apt-get install postfix mailutils (servidor + cliente)
  - nano /etc/postfix/main.cf
- Configurações acrescentadas:
- myhostname = mail.banc1.pgre
  - mydomain = banc1.pgre
  - mynetworks = 127.0.0.0/8, 192.168.1.0/24, 172.16.1.0/24
  - inet\_protocols = ipv4
  - home\_mailbox = Maildir/
- systemctl restart postfix
  - systemctl enable postfix
  - echo 'conteudo' | mail -s 'assunto' root /\* Comando para enviar mail \*/

Apesar de ser um mail aparentemente local, ou seja, enviado de um tux para o mesmo, obtivemos o respetivo tráfego SMTP desejado.



```
Computer
CPU Frequency: 2,00
CPU Usage: 1%
RAM Usage: 485M/7
Swap Usage: 160K/1B
Processes: 165 Running

Terminal
File Edit View Search Terminal Help
tux11:~# echo 'conteudo' | mail -s 'assunto' root
tux11:~#
```

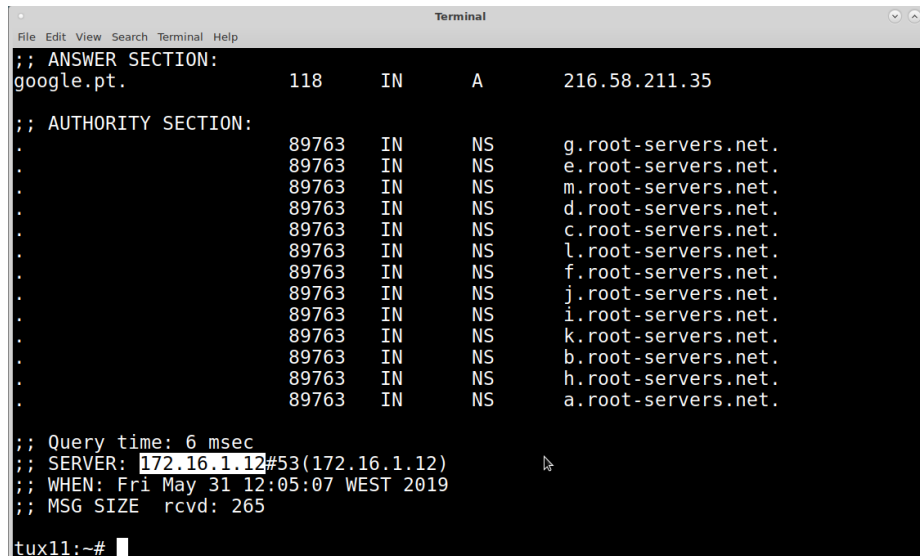
```
netstat
/var/mail
Save
1 From root@banc1.pgre Fri May 31 12:04:34 2019
2 Return-Path: <root@banc1.pgre>
3 X-Original-To: root
4 Delivered-To: root@banc1.pgre
5 Received: by tux11.netlab.fe.up.pt (Postfix, from userid 0)
6 id 5611A1A0468; Fri, 31 May 2019 12:04:34 +0100 (WEST)
7 To: root@banc1.pgre
8 Subject: assunto
9 MIME-Version: 1.0
10 Content-Type: text/plain; charset="UTF-8"
11 Content-Transfer-Encoding: 8bit
12 Message-Id: <20190531110434.5611A1A0468@tux11.netlab.fe.up.pt>
13 Date: Fri, 31 May 2019 12:04:34 +0100 (WEST)
14 From: root@banc1.pgre (root)
15
16 conteudo
17
```

Plain Text Tab Width: 8 Ln 1, Col 1 INS

## Servidor DNS

Para o servidor DNS executamos os comandos definidos no ficheiro também em anexo por ser demasiado extenso (dns\_config.txt).

Pode ver-se embaixo o pedido feito a partir do tux11 ao servidor alocado no tux12:



```
;; ANSWER SECTION:
google.pt.      118      IN      A       216.58.211.35

;; AUTHORITY SECTION:
.               89763    IN      NS      g.root-servers.net.
.               89763    IN      NS      e.root-servers.net.
.               89763    IN      NS      m.root-servers.net.
.               89763    IN      NS      d.root-servers.net.
.               89763    IN      NS      c.root-servers.net.
.               89763    IN      NS      l.root-servers.net.
.               89763    IN      NS      f.root-servers.net.
.               89763    IN      NS      j.root-servers.net.
.               89763    IN      NS      i.root-servers.net.
.               89763    IN      NS      k.root-servers.net.
.               89763    IN      NS      b.root-servers.net.
.               89763    IN      NS      h.root-servers.net.
.               89763    IN      NS      a.root-servers.net.

;; Query time: 6 msec
;; SERVER: 172.16.1.12#53(172.16.1.12)
;; WHEN: Fri May 31 12:05:07 WEST 2019
;; MSG SIZE rcvd: 265

tux11:~#
```

## Geração de tráfego

Finalmente utilizámos a ferramenta crontab para automatizar a geração de tráfego aos vários servidores criados:

#DNS

```
*/15 * * * * dig google.pt > /dev/null 2>&1
```

#NTP

```
*/10 * * * * ntpdate 172.16.1.13 > /dev/null 2>&1
```

#FTP

```
*/8 * * * * python /root/Desktop/ftp_python.py > /dev/null 2>&1
```

#Apache

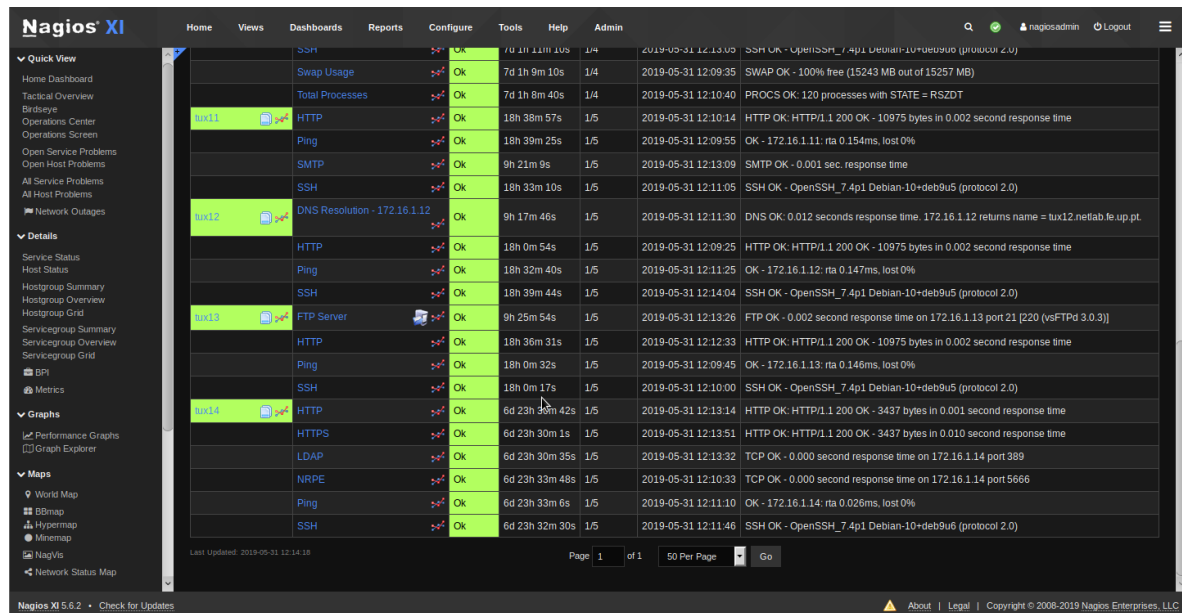
```
*/5 * * * * curl http://172.16.1.13 > /dev/null 2>&1
```

#Postfix

```
*/20 * * * * echo 'conteudo' | mail -s 'assunto' root > /dev/null 2>&1
```

# Nagios

O Nagios foi instalado e configurado automaticamente através de uma script disponível no site oficial do Nagios.



The screenshot displays the Nagios XI web interface. The top navigation bar includes links for Home, Views, Dashboards, Reports, Configure, Tools, Help, and Admin. A left sidebar contains a 'Quick View' section with links to Home Dashboard, Tactical Overview, Birdseye, Operations Center, Operations Screen, Open Service Problems, Open Host Problems, All Service Problems, All Host Problems, and Network Outages. Below this is a 'Details' section with links to Service Status, Host Status, Hostgroup Summary, Hostgroup Overview, Hostgroup Grid, Servicegroup Summary, Servicegroup Overview, Servicegroup Grid, BPI, Metrics, and Graphs. The main content area shows a list of services for four hosts: tux11, tux12, tux13, and tux14. Each host has a status icon (green for OK, red for CRITICAL) and a list of services with their status, last update time, and details. For example, tux11 has services like SSH, Swap Usage, Total Processes, HTTP, Ping, SMTP, and DNS Resolution. The bottom of the interface shows a footer with 'Nagios XI 5.6.2' and a 'Check for Updates' link.

Host	Service	Status	Last Update	Details
tux11	SSH	OK	2019-05-31 12:13:05	SSH OK - OpenSSH_7.4p1 Debian-10+deb9u5 (protocol 2.0)
tux11	Swap Usage	OK	2019-05-31 12:09:35	SWAP OK - 100% free (15243 MB out of 15257 MB)
tux11	Total Processes	OK	2019-05-31 12:10:40	PROCS OK: 120 processes with STATE = RSZDT
tux11	HTTP	OK	2019-05-31 12:10:14	HTTP OK: HTTP/1.1 200 OK - 10975 bytes in 0.002 second response time
tux11	Ping	OK	2019-05-31 12:09:55	OK - 172.16.1.11: rta 0.154ms, lost 0%
tux11	SMTP	OK	2019-05-31 12:13:09	SMTP OK - 0.001 sec. response time
tux11	SSH	OK	2019-05-31 12:11:05	SSH OK - OpenSSH_7.4p1 Debian-10+deb9u5 (protocol 2.0)
tux12	DNS Resolution - 172.16.1.12	OK	2019-05-31 12:11:30	DNS OK: 0.012 seconds response time. 172.16.1.12 returns name = tux12.netlab.le.up.pt.
tux12	HTTP	OK	2019-05-31 12:09:25	HTTP OK: HTTP/1.1 200 OK - 10975 bytes in 0.002 second response time
tux12	Ping	OK	2019-05-31 12:11:25	OK - 172.16.1.12: rta 0.147ms, lost 0%
tux12	SSH	OK	2019-05-31 12:14:04	SSH OK - OpenSSH_7.4p1 Debian-10+deb9u5 (protocol 2.0)
tux13	FTP Server	OK	2019-05-31 12:13:26	FTP OK - 0.002 second response time on 172.16.1.13 port 21 [220 (vsFTPd 3.0.3)]
tux13	HTTP	OK	2019-05-31 12:12:33	HTTP OK: HTTP/1.1 200 OK - 10975 bytes in 0.002 second response time
tux13	Ping	OK	2019-05-31 12:09:45	OK - 172.16.1.13: rta 0.146ms, lost 0%
tux13	SSH	OK	2019-05-31 12:10:00	SSH OK - OpenSSH_7.4p1 Debian-10+deb9u5 (protocol 2.0)
tux14	HTTP	OK	2019-05-31 12:13:14	HTTP OK: HTTP/1.1 200 OK - 3437 bytes in 0.001 second response time
tux14	HTTPS	OK	2019-05-31 12:13:51	HTTP OK: HTTP/1.1 200 OK - 3437 bytes in 0.010 second response time
tux14	LDAP	OK	2019-05-31 12:13:32	TCP OK - 0.000 second response time on 172.16.1.14 port 389
tux14	NRPE	OK	2019-05-31 12:10:33	TCP OK - 0.000 second response time on 172.16.1.14 port 5666
tux14	Ping	OK	2019-05-31 12:11:10	OK - 172.16.1.14: rta 0.026ms, lost 0%
tux14	SSH	OK	2019-05-31 12:11:46	SSH OK - OpenSSH_7.4p1 Debian-10+deb9u5 (protocol 2.0)

# Zabbix

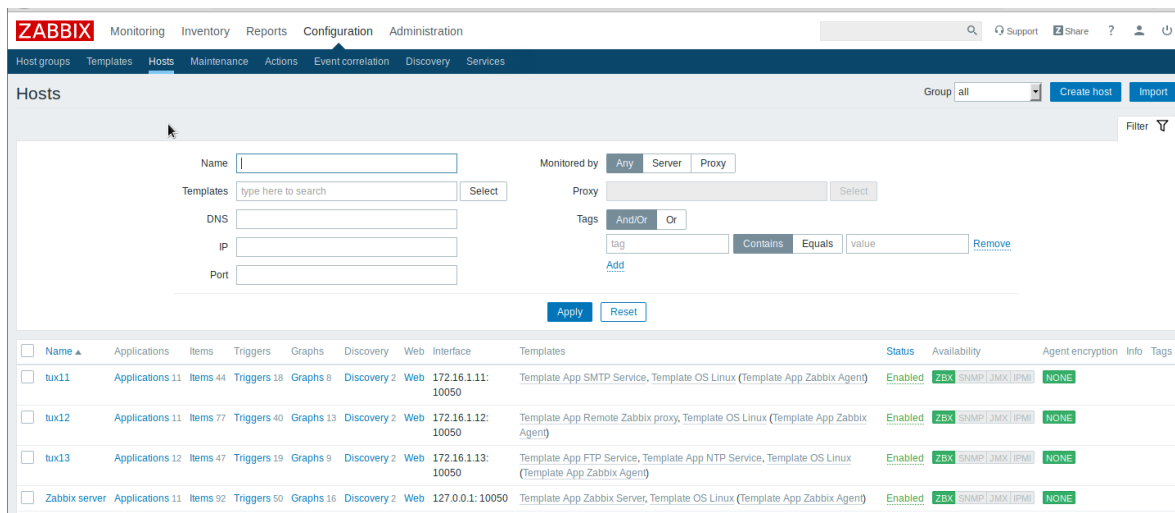
O Zabbix por outro lado, foi mais complicado de instalar e configurar. Para tal foram seguidos os passos que se encontram no site oficial do Zabbix e foram criados três clientes em três tux's e um servidor no restante.

Para tal usamos os seguintes comandos para configurar o servidor:

- `wget https://repo.zabbix.com/zabbix/4.2/debian/pool/main/z/zabbix-release/zabbix-release_4.2-1+stretch_all.deb`
  - `dpkg -i zabbix-release_4.2-1+stretch_all.deb`
  - `apt update`
  - `apt install zabbix-server-mysql zabbix-frontend-php zabbix-agent`
  - `mysql -uroot -p`  
password /\*Inserir nova password\*/  
`mysql> create database zabbix character set utf8 collate utf8_bin;`  
`mysql> grant all privileges on zabbix.* to zabbix@localhost identified by 'password';`  
`mysql> quit;`
  - `zcat /usr/share/doc/zabbix-server-mysql*/create.sql.gz | mysql -uzabbix -p zabbix`
  - `nano /etc/zabbix/zabbix_server.conf`
- Configurações acrescentadas:

- DBPassword=password /\*password = password pretendida”
- systemctl restart zabbix-server zabbix-agent apache2
- systemctl enable zabbix-server zabbix-agent apache2

No cliente instalamos apenas o pacote zabbix-agent e após definido o IP do servidor ( 172.16.1.14 ), criámos os hosts na interface gráfica e associamos os respetivos templates:



## Resultados Obtidos

Nas seguintes imagens é possível verificar os resultados obtidos, indicadores de um bom funcionamento de ambas as ferramentas.

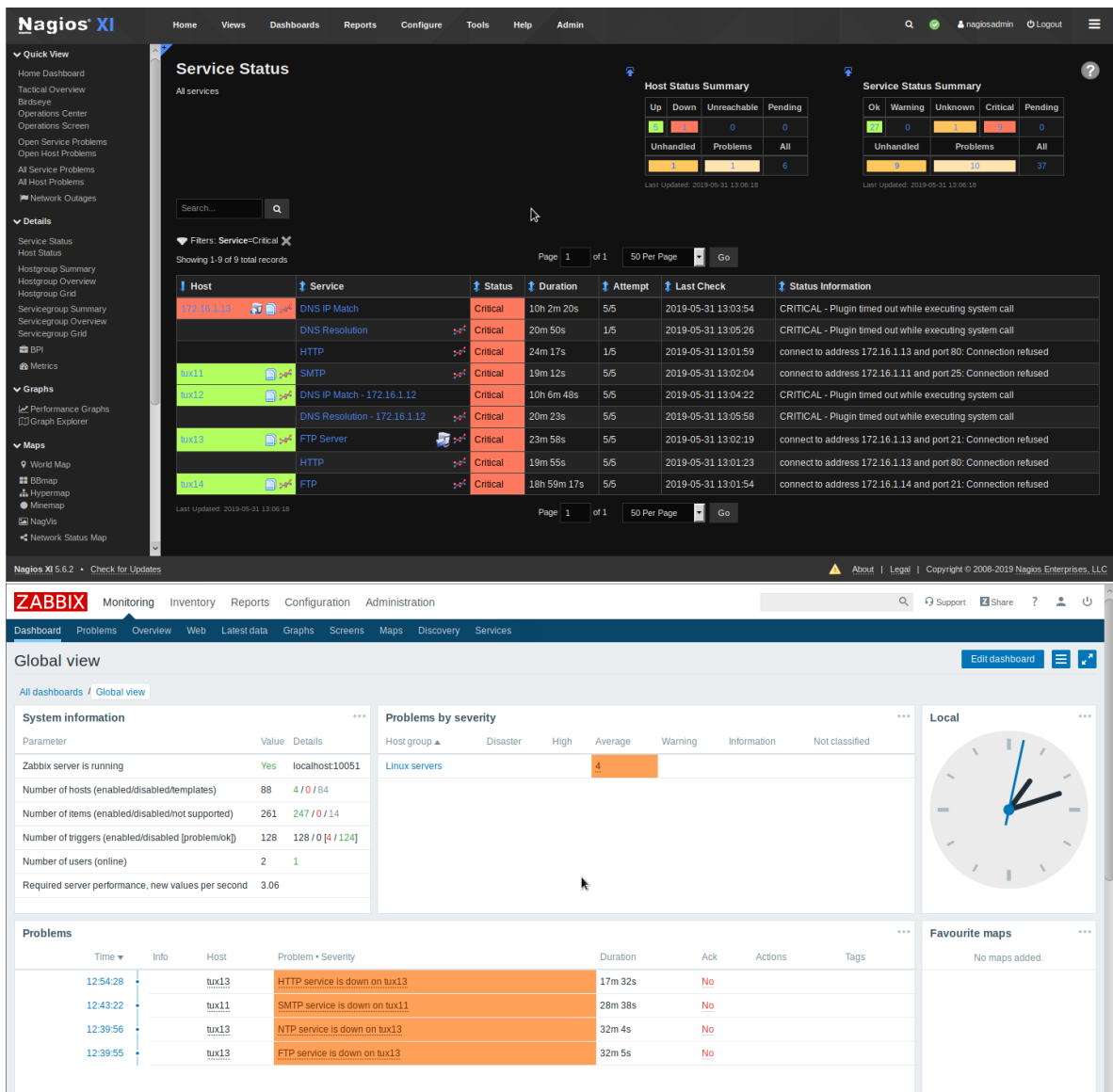
Zabbix:



Posteriormente, foram provocadas falhas nos servidores, de modo a verificar o que as duas ferramentas apresentavam:

Estes erros foram originados ao desativar o daemon com o comando `service * stop` e logo pudemos observar em ambas as interfaces os avisos de que algo estaria a funcionar mal.





Devido a uma falta de template de associação ao protocolo DNS não foi possível ao zabbix identificar a falha neste servidor.

## Comparação de ferramentas e conclusões finais

Após algumas horas a trabalhar com ambas as ferramentas podemos concluir que são ambas poderosas à sua maneira e ao mesmo tempo diferentes.

O software que o Nagios disponibiliza no website oficial facilita imenso a configuração deste, e a interface gráfica é acessível. Associando os serviços a cada um dos hosts rapidamente obtivemos informação acerca do tráfego a que estes estavam sujeitos.

Já o zabbix não foi tão simples de configurar pois não havia um script que fazia todo o trabalho por nós, mas com uma pesquisa razoável pela web rapidamente se percebeu o funcionamento da ferramenta. De notar que o zabbix oferece informação acerca da utilização de memória e CPU, coisa que o nagios não demonstra. Existem também métodos disponíveis que não utilizámos como os alertas por e-mail, encriptação de dados e uma quantia enorme de templates para utilizar (inclusive feitos em desenvolvimento externo ao software oficial), este último tanto no zabbix como no nagios.

## Ferramentas Grafana e openDCIM

As ferramentas Nagios e Zabbix são ferramentas mais direccionadas à monitorização do tráfego numa rede. Estas conseguem detetar quando ocorre uma falha no sistema assim como quando essa falha é resolvida.

Por outro lado, as ferramentas Grafana e openDCIM são utilizadas para analisar os logs.

Estas ferramentas têm características únicas pelo que, o uso de várias pode ser benéfico na análise e monitorização da rede.

## Fontes

<https://www.zabbix.com>

<https://www.zabbix.com/download>

<https://www.nagios.org>

<https://www.nagios.com/downloads/nagios-xi/linux/>

<https://www.digitalocean.com/community/tutorials/how-to-configure-bind-as-a-private-network-dns-server-on-ubuntu-14-04>

<https://grafana.com>

<https://opendcim.org>