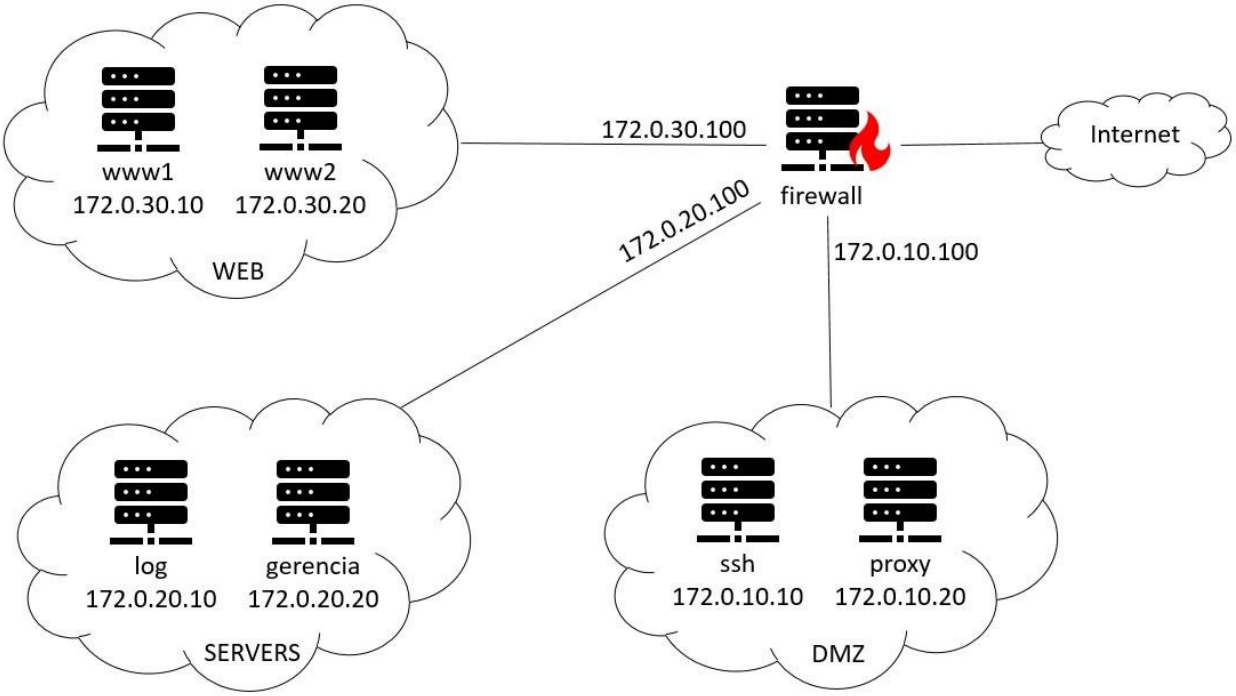
Projeto Final

INF574 – Administração e Segurança

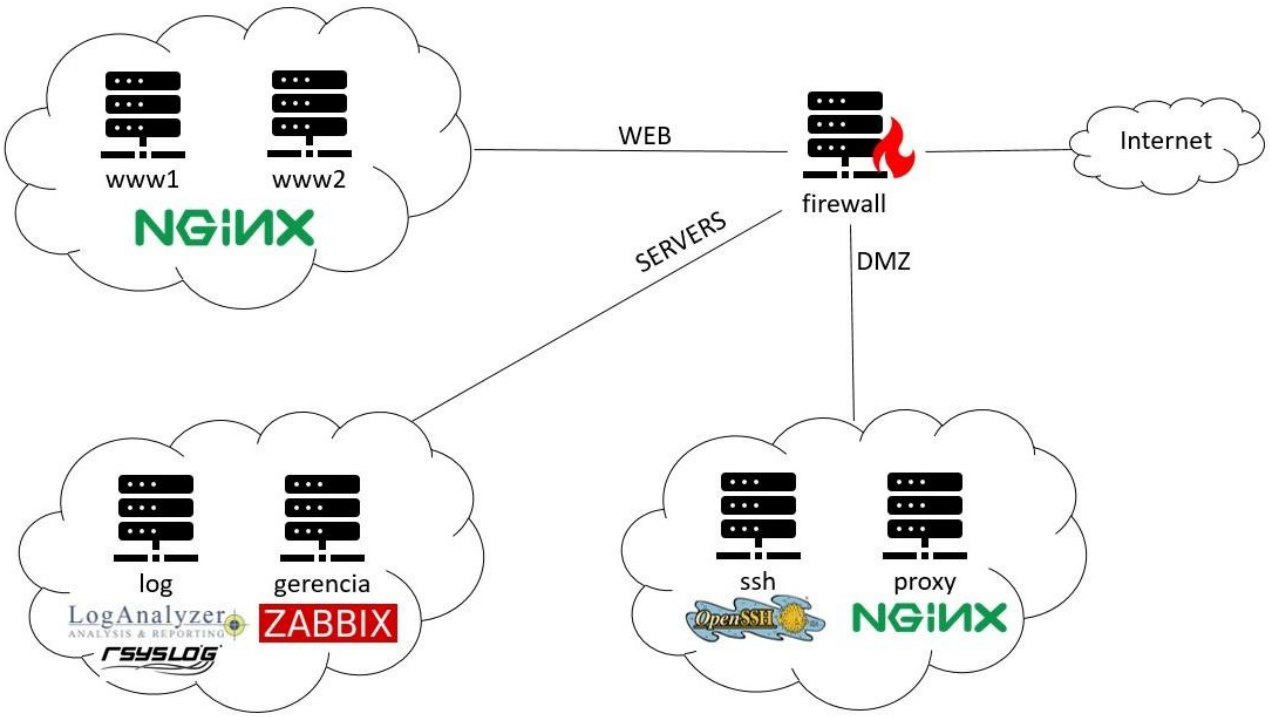
UNICAMP – 2019

Gustavo P de O Celani

Topologia



Ferramentas utilizadas em cada nó



Firewall

O firewall é o elemento de borda da topologia de rede no qual possui o papel de controlar o roteamento da rede. Ele garante que apenas a rede DMZ possua acesso à internet e as demais redes apenas são visíveis entre si.

Além desta restrição também deve ser configurado apenas a liberação das portas 4578 e 443, usadas, respectivamente, para SSH e HTTPS.

Para possibilitar acesso à internet pela rede DMZ (172.0.10.0/24) foi realizado um NAT:

$ iptables -t nat -A POSTROUTING --source 172.0.10.0/24 --out-interface eth0 -j MASQUERADE

Exemplo de regras de firewall:

Ação	Container	IP	Porta
Permitir	SSH	172.0.10.10	4578
Permitir	Proxy	172.0.10.20	443

SSH

Este container faz parte da rede DMZ e tem acesso à internet de acordo com as regras de firewall. Ele é responsável por acessar todas as máquinas da topologia via SSH. Sendo a única interface de comunicação possível para acesso.

Configurações de hardening:

- Protocolo SSHv2
- Desabilitando login como root
- Sessões controladas
    - Client Alive Interval: 300
    - Client Alive Count Max: 3
- Evitando a porta padrão (22 -> 4578)
- Desabilitando encaminhamento X11
- Apenas o usuário “ssh_user” está habilitado para ser logado
- 3-Factor Authentication
    - Password
    - RSA Key
    - Challenge PAM with Google-Authenticator
- Fail2ban
    - Selected ban time
    - 3 Max retry
    - Alias to watch logs: `$ log-fail2ban`
- Firewall rules

Conexão no container SSH (3 fatores de autenticação):

$ ssh -i ./conf/ssh/ssh_ssh_key ssh_user@172.0.10.10 -p 4578

```
┌─[root@burton-pc]─[master*]─[/home/burton/git_projects/shell_scripts/lxd/monitored_environment]
└──╼  # ssh -i ./conf/ssh/ssh_ssh_key ssh_user@172.0.10.10 -p 4578
Password:
Verification code:
Linux ssh 5.0.0-27-generic #28~18.04.1-Ubuntu SMP Thu Aug 22 03:00:32 UTC 2019 x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Nov 15 15:48:06 2019 from 172.0.10.1
ssh_user@ssh:~$
ssh_user@ssh:~$
```

Uma vez logado no container SSH é possível utilizar diferentes alias criados para facilitar e abstrair as trocas de chave SSH para todas as máquinas presentes na rede:

```
# Firewall
$ ssh-firewall

# Network DMZ
$ ssh-ssh
$ ssh-proxy

# Network SERVERS
$ ssh-log
$ ssh-gerencia

# Network WEB
$ ssh-www1
$ ssh-www2
```

```
ssh_user@ssh:~$
ssh_user@ssh:~$ ssh-www1
Linux www1 5.0.0-27-generic #28~18.04.1-Ubuntu SMP Thu Aug 22 03:00:32 UTC 2019 x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Nov 15 15:48:32 2019 from 172.0.10.10
www1_user@www1:~$
www1_user@www1:~$
```

Adição do Google-Authenticator no arquivo ***sshd***:

```
# Enabling 2 factors authentication
auth required pam_google_authenticator.so
```

Configurações personalizadas no arquivo ***sshd_config***:

```
# Enabling only Protocol 2
Protocol 2

# Disabling Root Login
PermitRootLogin no

# Setting up client sessions
ClientAliveInterval 300
ClientAliveCountMax 3

# Avoiding default port (22)
Port 4578

# Disabling password authentication
PasswordAuthentication no

# Disabling X11
X11Forwarding no

# Allowed users
AllowUsers ssh_user

# Key authentication
RSAAuthentication yes
PubkeyAuthentication yes
AuthorizedKeysFile /ssh_user/.ssh/authorized_keys

# Setting up Google-Authenticator
UsePAM yes
ChallengeResponseAuthentication yes
AuthenticationMethods publickey,password publickey,keyboard-interactive
```

Configurações personalizadas no arquivo ***jail.local***:

```
# "bantime" is the number of seconds that a host is banned.
bantime  = 10

# A host is banned if it has generated "maxretry" during the last "findtime"
findtime  = 10

# "maxretry" is the number of failures before a host get banned.
maxretry = 3
```

Configurações personalizadas no arquivo ***jail.local***:

```
# Option: logtarget
# Notes.: Set the log target. This could be a file, SYSLOG, STDERR or STDOUT.
#         Only one log target can be specified.
#         If you change logtarget from the default value and you are
#         using logrotate -- also adjust or disable rotation in the
#         corresponding configuration file
#         (e.g. /etc/logrotate.d/fail2ban on Debian systems)
# Values: [ STDOUT | STDERR | SYSLOG | FILE ]  Default: STDERR
#
logtarget = /var/log/fail2ban.log
```

Configurações personalizadas no arquivo ***.bashrc***:

```
# SSH Alias - Network DMZ
alias ssh-ssh='ssh -i /ssh_user/.ssh/ssh_ssh_key ssh_user@172.0.10.10 -p 4578'
alias ssh-proxy='ssh -i /ssh_user/.ssh/ssh_ssh_key proxy_user@172.0.10.20 -p
4578'

# SSH Alias - Network SERVERS
alias ssh-log='ssh -i /ssh_user/.ssh/ssh_ssh_key log_user@172.0.20.10 -p 4578'
alias ssh-gerencia='ssh -i /ssh_user/.ssh/ssh_ssh_key gerencia_user@172.0.20.20
-p 4578'

# SSH Alias - Network WEB
alias ssh-www1='ssh -i /ssh_user/.ssh/ssh_ssh_key www1_user@172.0.30.10 -p 4578'
alias ssh-www2='ssh -i /ssh_user/.ssh/ssh_ssh_key www2_user@172.0.30.20 -p 4578'

# fail2ban
alias log-fail2ban='tail -f /var/log/fail2ban.log'
```

Proxy

Este container faz parte da rede DMZ e tem acesso à internet de acordo com as regras de firewall. Ele é responsável por prover um proxy reverso HTTPS atuando também como balanceador de carga utilizando Nginx.

Configurações de hardening:

- TLS over HTTP (HTTPS)
- Load Balancer for WWW1 and WWW2
- Reverse Proxy
- DDoS Prevention
- Firewall rules

Os serviços disponíveis pelo proxy são:

- Load Balancer of web servers
  - ***https://172.0.10.20***
  - ***https://172.0.10.20/www***

- WWW1 Web Server
  - ***https://172.0.10.20/www1***

- WWW2 Web Server
  - ***https://172.0.10.20/www2***

- LogAnalyzer
  - ***https://172.0.10.20/log***

- Zabbix
  - ***https://172.0.10.20/gerencia***

Configurações personalizadas no arquivo ***default*** na configuração do Nginx:

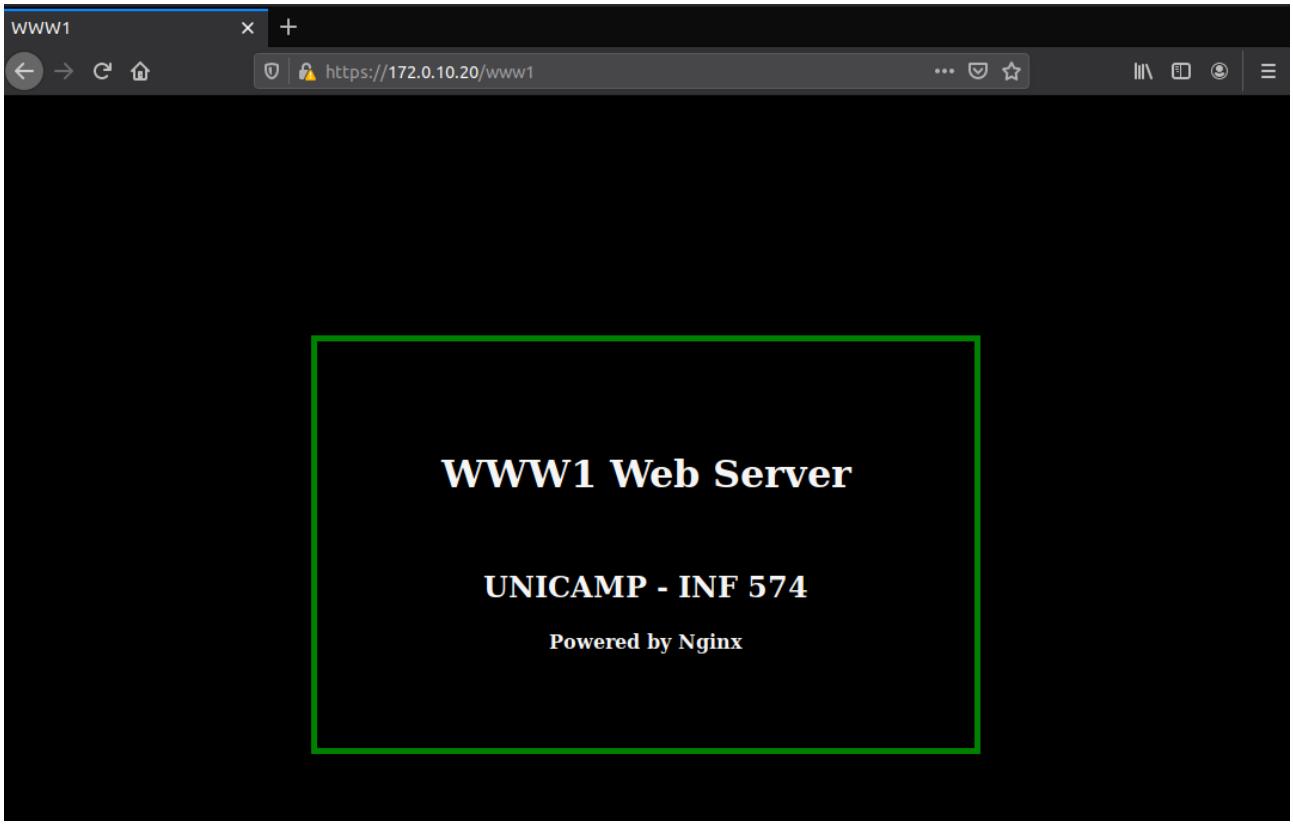
```
#
# Reverse HTTPS Proxy with Load Balancer Configuration
#

upstream load_balancer {
    # WWW1 Web Server
    server 172.0.30.10;

    # WWW2 Web Server
    server 172.0.30.20;
}


server {

    # Server Name
    server_name proxy;

    # Listen port 443 using HTTPS Protocol
    listen 443 ssl;

    # SSL CA Certificate
    ssl_certificate /etc/nginx/ssl/proxy_nginx.crt;

    # SSL Key
    ssl_certificate_key /etc/nginx/ssl/proxy_nginx.key;

    # Load Balancer using 'load_balancer' upstream
    location / {
        proxy_pass http://load_balancer;
    }

    # Load Balancer using 'load_balancer' upstream
    location /www {
        proxy_pass https://load_balancer;
    }

    # Reverse proxy for WWW1 Web Server
    location /www1 {
        proxy_pass http://172.0.30.10/;
    }

    # Reverse proxy for WWW2 Web Server
    location /www2 {
        proxy_pass http://172.0.30.20/;
    }

    # Reverse proxy for Loganalyzer Server
    location /log {
        proxy_pass http://172.0.20.10/loganalyzer;
    }

    # Reverse proxy for Zabbix Server
    location /gerencia {
        proxy_pass http://172.0.20.20/zabbix;
    }
}
```

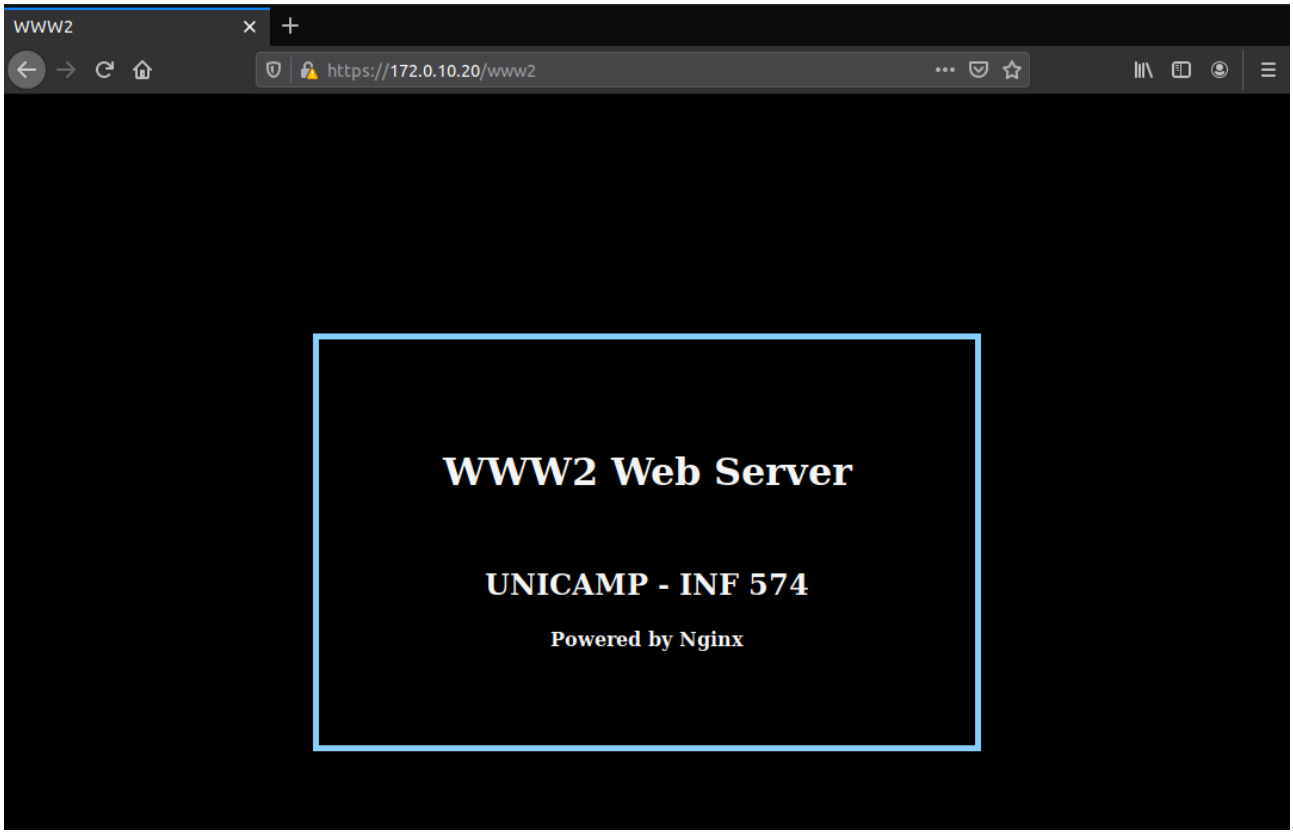
Certificado x509 auto-assinado gerado para TLS:

```
┌─[burton@burton-pc]─[master*]─[~/git_projects/shell_scripts/lxd/monitored_environment/conf/proxy]
└──╼ $ openssl x509 -in proxy_nginx.crt -text -noout
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            5d:db:97:51:ed:43:eb:08:17:bd:a5:d0:48:bc:c2:e4:e6:96:06:5c
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C = BR, ST = Sao Paulo, L = Campinas, O = UNICAMP, OU = INF574, CN = PROXY_CA, emailAddress = proxy@ma
il.com
        Validity
            Not Before: Nov 15 00:53:41 2019 GMT
            Not After : Nov 14 00:53:41 2020 GMT
        Subject: C = BR, ST = Sao Paulo, L = Campinas, O = UNICAMP, OU = INF574, CN = PROXY_CA, emailAddress = proxy@m
ail.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                RSA Public-Key: (2048 bit)
                Modulus:
                    00:d2:0d:bc:e0:ec:38:52:6c:15:c6:c3:94:b9:6f:
                    3a:d6:a5:1e:e0:66:31:2a:fc:03:34:31:8a:42:bc:
                    48:9e:29:c4:e4:d0:d4:ad:36:c3:1c:0c:c0:7e:4b:
                    65:a8:9f:03:03:85:f3:7a:3c:34:4b:9e:58:e8:3d:
                    8c:86:ac:29:29:c8:eb:09:7b:33:4e:f1:62:f3:23:
                    a6:64:97:e6:f2:13:93:40:95:4d:91:8e:10:2e:4e:
                    c7:59:1d:06:b5:94:23:fe:6f:4e:b0:7e:6c:ef:40:
                    28:df:23:58:58:a9:bf:70:8c:78:83:9c:b0:4a:0c:
                    66:57:a0:59:b8:5c:3f:45:06:69:27:8e:10:66:e0:
                    a6:86:33:b3:9a:4e:07:59:53:5b:26:f3:0e:06:ae:
                    1b:18:d1:b2:5f:32:bb:66:52:14:97:51:7f:35:4b:
                    74:5d:17:cd:f6:69:01:e4:7c:24:f1:73:f9:1b:53:
                    ef:0a:80:b0:e3:22:06:ec:f6:0b:d7:2d:df:06:4c:
                    e6:a8:ac:3d:48:05:82:c7:3c:e5:ae:a8:16:77:63:
                    b0:12:3e:b6:a3:a0:58:56:d1:37:0f:54:76:e4:0a:
                    10:a7:77:84:c1:bc:00:d8:24:fb:25:c0:18:a2:14:
                    37:b8:5f:a5:db:53:ce:86:45:bd:58:34:82:0a:7e:
                    fd:15
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Subject Key Identifier:
                60:2D:78:CA:01:27:4D:04:70:54:B2:2A:07:4E:C4:41:79:B8:D4:2B
            X509v3 Authority Key Identifier:
                keyid:60:2D:78:CA:01:27:4D:04:70:54:B2:2A:07:4E:C4:41:79:B8:D4:2B

            X509v3 Basic Constraints: critical
                CA:TRUE
    Signature Algorithm: sha256WithRSAEncryption
         9c:10:1d:80:52:26:7d:c1:de:98:eb:99:ed:f0:a3:01:c3:f5:
         39:da:c6:9a:ff:cc:22:86:ac:61:5d:f9:7c:a0:a6:00:82:c9:
         33:39:00:7a:5a:cc:20:45:98:c4:f4:09:b6:1c:9a:38:fa:85:
         97:89:12:8a:3a:d7:df:82:b3:13:b4:d3:de:bc:c7:5c:f5:89:
         a5:dd:9e:8a:13:a6:9a:75:5a:82:c9:16:74:6a:c9:a0:fa:45:
         36:5e:6e:da:4a:06:b7:7e:eb:d7:88:9a:b7:16:61:e9:52:3e:
         64:b1:88:a9:a7:de:e1:13:9f:1b:a9:4d:ba:bd:8b:6e:06:d8:
         9b:db:9e:26:81:fa:f4:1b:0e:c5:85:f3:60:13:6f:64:28:1d:
         90:7e:f3:4c:b8:03:ef:68:22:e0:7b:06:39:5f:8f:71:49:b0:
         13:13:1c:e4:c5:6a:6d:40:ec:74:ad:be:24:05:7d:58:8a:c6:
         70:7c:63:52:e2:e2:e3:66:43:36:43:ee:e0:d0:8a:9f:0c:ac:
         54:ea:1d:d4:86:1b:a2:4c:54:8f:f9:a9:36:72:1d:92:7b:09:
         eb:6b:09:a5:7d:01:70:14:ee:7d:70:3f:f2:3b:16:19:d9:70:
         0c:1e:49:24:5a:10:b5:88:7b:96:d8:6d:a4:4d:e8:fd:34:2a:
         e8:c5:c6:ce
┌─[burton@burton-pc]─[master*]─[~/git_projects/shell_scripts/lxd/monitored_environment/conf/proxy]
```

WWW1 e WWW2

Estes containers fazem parte da rede WEB e não têm acesso à internet de acordo com as regras de firewall. Cada um deles provê um servidor HTTP na porta 80 utilizando Nginx.





Configurações personalizadas no arquivo ***default*** na configuração do Nginx de WWW1:

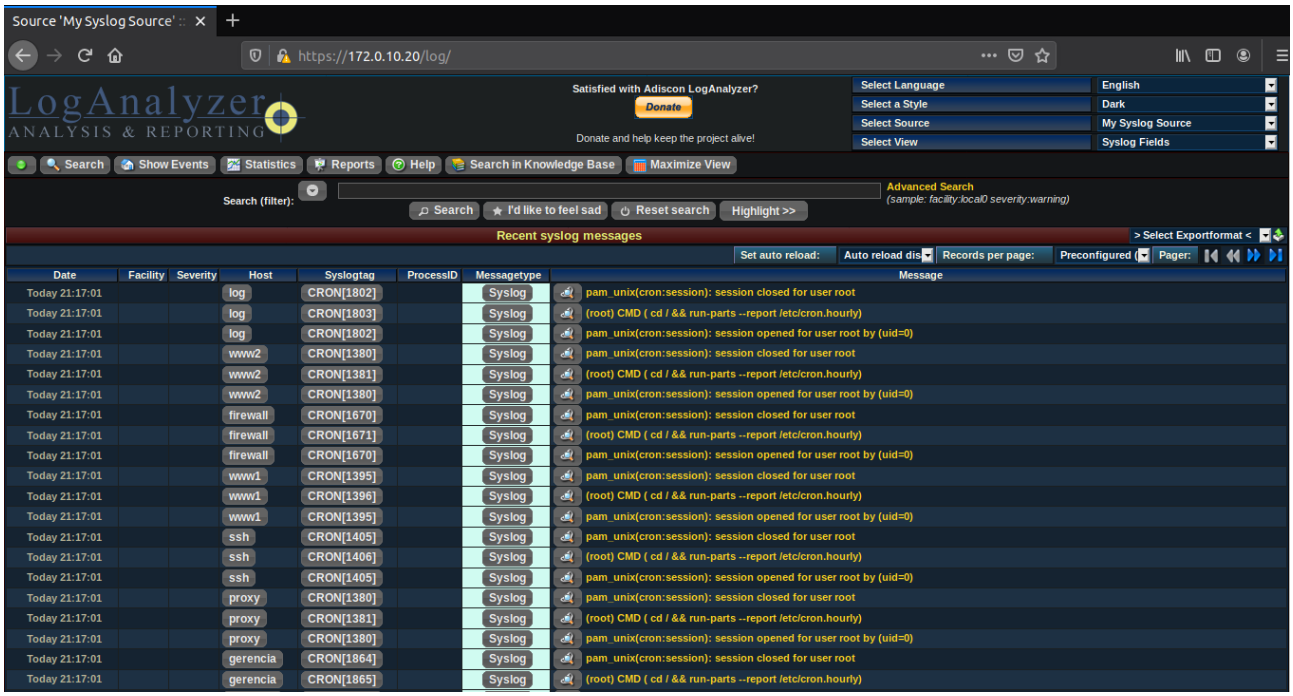
```
#
# WWW1 Server Configuration
#
server {

    # Listen port 80 using HTTP Protocol
    listen 80;
    listen [::]:80 default_server ipv6only=on;

    # Root files path
    root /usr/share/nginx/html;
    index index.html index.htm;

    # Server Name
    server_name www1_server;

    location / {
                # First attempt to serve request as file, then
                # as directory, then fall back to displaying a 404.
                try_files $uri $uri/ =404;
        }
}
```

Configurações personalizadas no arquivo ***default*** na configuração do Nginx de WWW2:

```
#
# WWW2 Server Configuration
#
server {

    # Listen port 80 using HTTP Protocol
    listen 80;
    listen [::]:80 default_server ipv6only=on;

    # Root files path
    root /usr/share/nginx/html;
    index index.html index.htm;

    # Server Name
    server_name www2_server;

    location / {
                # First attempt to serve request as file, then
                # as directory, then fall back to displaying a 404.
                try_files $uri $uri/ =404;
        }
}
```

Log Server

Este container faz parte da rede SERVERS e não têm acesso à internet de acordo com as regras de firewall. Ele provê um servidor de logs centralizado que coleta os logs por meio do Rsyslog e os exibe de forma gráfica utilizando LogAnalyzer.



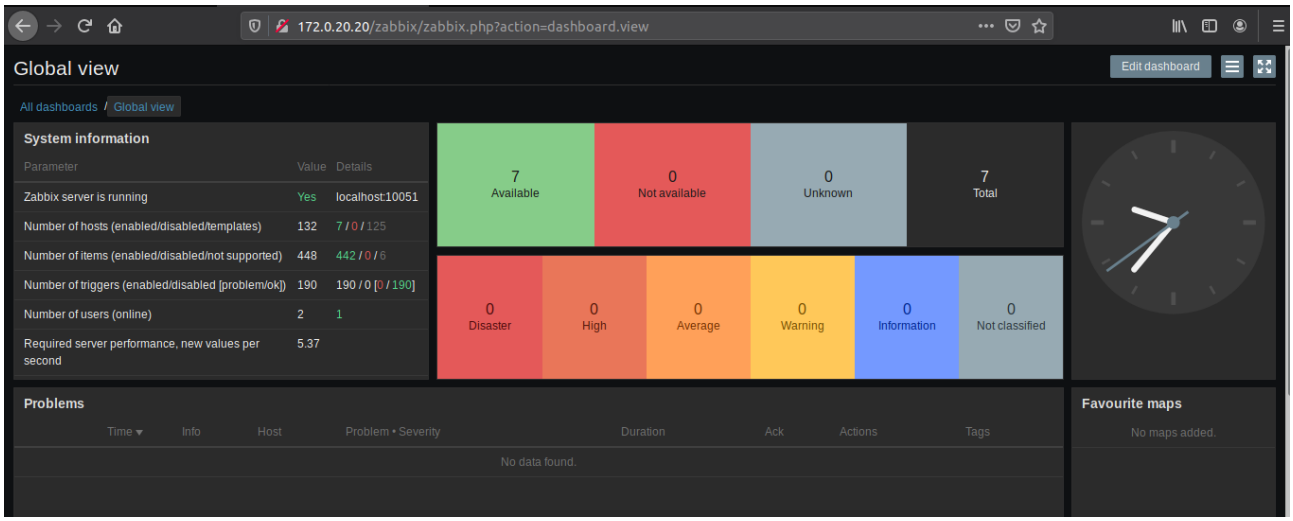
Configurações personalizadas no arquivo ***rsyslog.conf*** na configuração do Rsyslog server:

```
# provides support for local system logging
module(load="imuxsock")
# provides kernel logging support
module(load="imklog")

# Provides UDP syslog reception
module(load="imudp")
input(type="imudp" port="5689")
$AllowedSender UDP, 127.0.0.1, 172.0.10.0/24, 172.0.20.0/24, 172.0.30.0/24

# Provides TCP syslog reception
module(load="imtcp")
input(type="imtcp" port="5689")
$AllowedSender TCP, 127.0.0.1, 172.0.10.0/24, 172.0.20.0/24, 172.0.30.0/24

# Individual Log Template
# $template remote-incoming-logs,"/var/log/inf574/%HOSTNAME%/%PROGRAMNAME%.log"
# *.* ?remote-incoming-logs
# & ~

# General Log Template
$template remote-incoming-logs,"/var/log/inf574"
*.* ?remote-incoming-logs
& ~
```

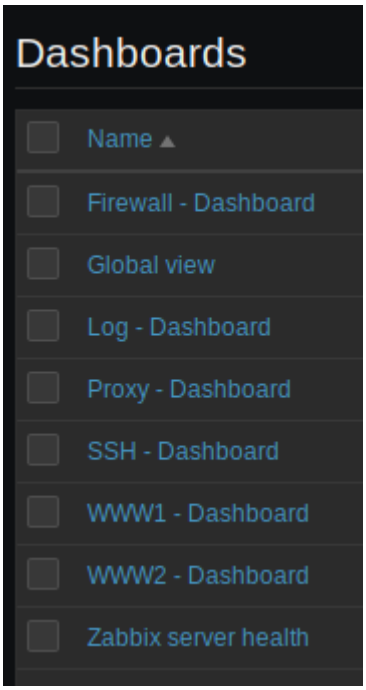
Gerencia Server

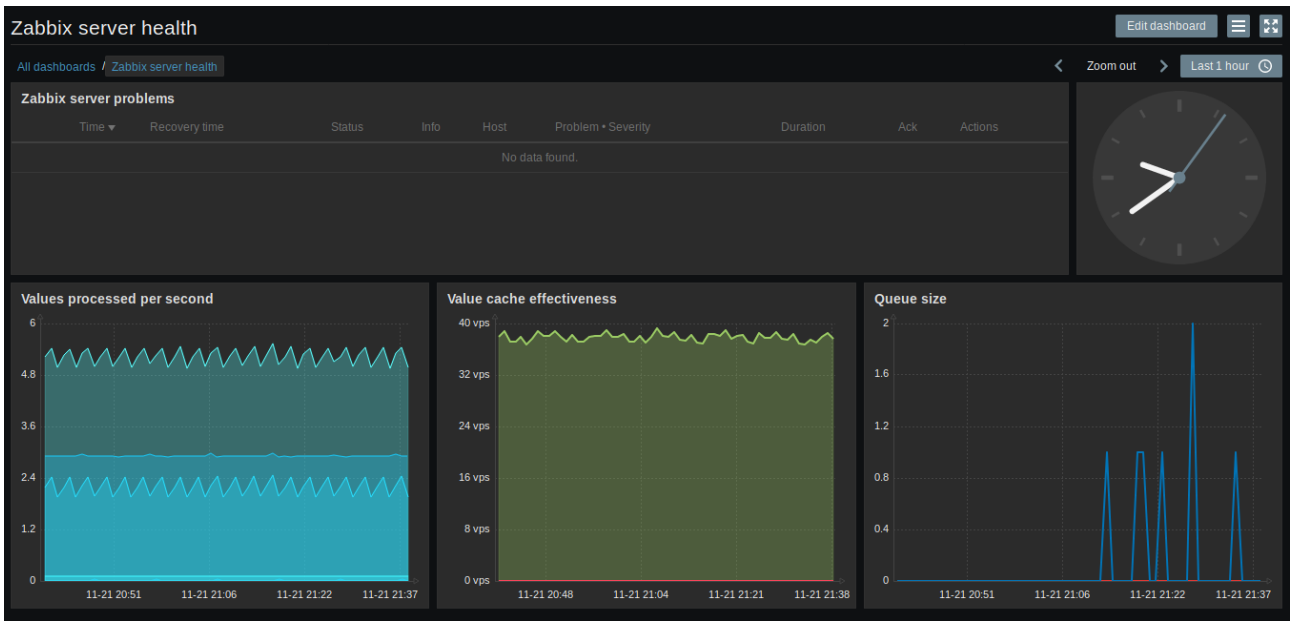
Este container faz parte da rede SERVERS e não têm acesso à internet de acordo com as regras de firewall. Ele provê um gerenciador centralizado de toda a rede utilizando Zabbix.

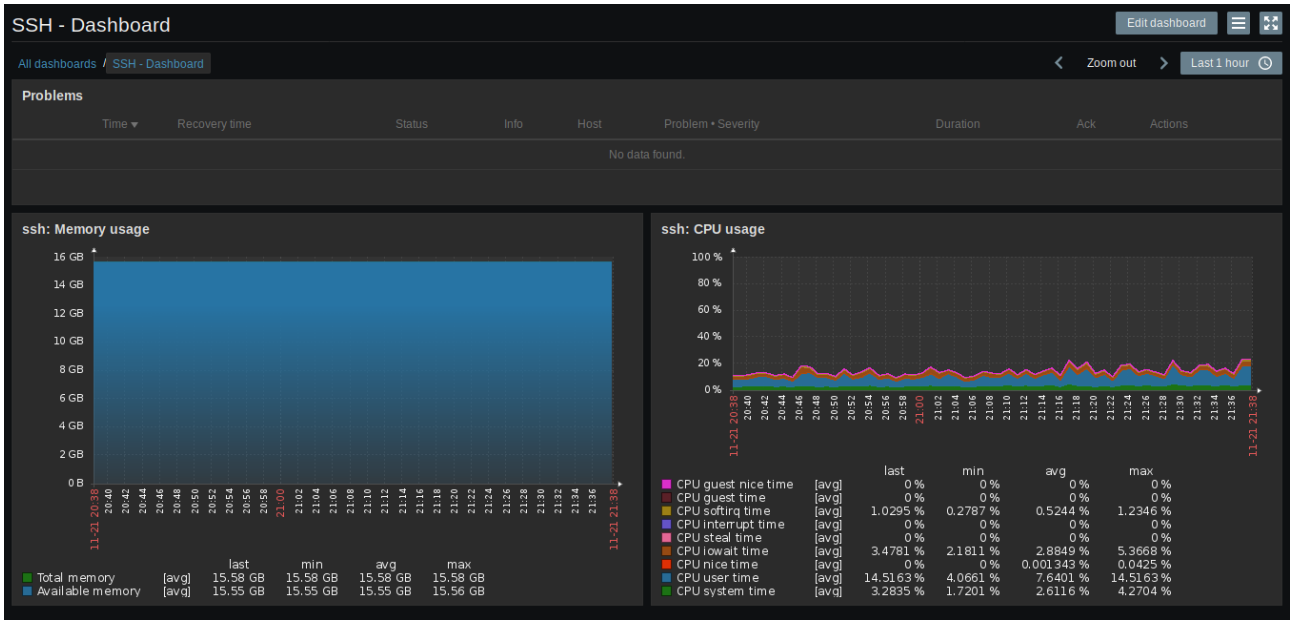
Todos os 7 containers presentes na topologia foram provisionados por meio do Zabbix agent.



Além do dashboard padrão, também foram criados um dashboard individual para acompanhar o estado de cada nó da rede:







Configurações personalizadas no arquivo ***zabbix_server.conf*** na configuração do Zabbix:

```
DBName=zabbix
DBUser=zabbix
DBPassword=password
```

Configurações personalizadas no arquivo ***zabbix_agent.conf*** na configuração do Zabbix:

```
Server=172.0.20.20
Hostname=Zabbix server
```