

LICENCIATURA EM SEGURANÇA INFORMÁTICA EM REDES DE COMPUTADORES

SISTEMAS CRITICOS

Trabalho Prático 1

Cluster altamente disponível, tolerante a falhas e com
balanceamento de carga

Realizado por:
Francisco Sousa Nº8200397

Conteúdo

Introdução	4
Cenário	5
Máquina Proxy	6
Instalação dos serviços:.....	6
Configuração HAproxy	6
Configuracao Web Cluster	6
Autenticação dos nós:	6
PROXY1:.....	6
PROXY2:.....	6
Criação e inicialização do cluster:	7
Propriedades do cluster:	7
Criação, adição e configuração dos recursos:	7
VirtualIP	7
HAProxy.....	8
Configurações adicionais:.....	8
Status do cluster.....	8
Máquinas Web	9
Criação e Configuração do RAID 1.....	9
Configuração Nginx:	12
Configuracao GlusterFS	12
Instalação:	12
Autenticação das máquinas:	13
Criação do volume e configuração:.....	13
Site da empresa.....	14
Ferramenta Cacti.....	14
Instalação:	14
Configuração MariaDB:	15
Instalação e configuração do Cacti:.....	16
Ferramenta Ganglia:.....	18
Instalação:	18
Instalação do ganglia:.....	19
Instalar ganglia nos clientes:	20
Configuração máquinas SQL.....	21
Instalação MariaDB	21

Configuração MariaDB	22
Configuração SQL Cluster	24
Instalação:	24
Autenticação dos nós:	24
Criação do cluster:	24
Inicialização do cluster:	24
Propriedades do cluster:	24
Criação, adição e configuração de recursos:	25
Configuração do gluster	26
Criação e inicialização do volume:	26
Montagem de volumes:	26
Ferramenta para efetuar x números de pedidos por segundo	27
Apache JMeter	27
Instalação:	27
Configuração	27
Avaliar o impacto que estes pedidos têm na memória RAM e no CPU	29
1 Servidor	29
2 Servidores	29
Análise qualitativa desses resultados	30
Balanceamento com 2 servidores:	30
Inexistência de balanceamento (1 servidor)	31
Conclusão	32
Anexos	33

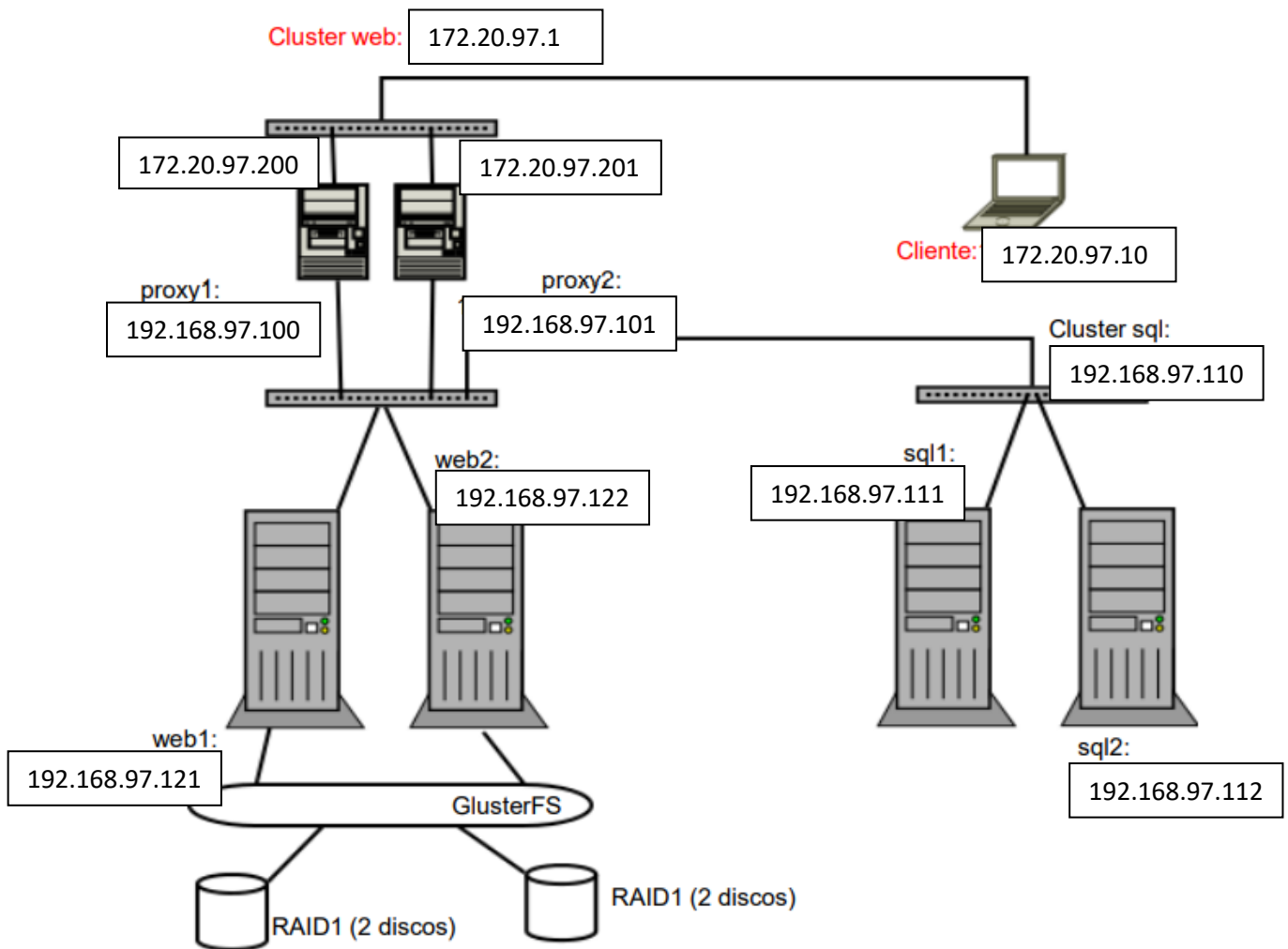
Introdução

No âmbito da unidade curricular de Sistemas Críticos, foi-me proposto a realização de um trabalho prático, com o objetivo de realizar a instalação de um mini-cluster, para alojamento de páginas de negócio de uma empresa, onde se pertence ter uma infraestrutura de alta disponibilidade, tolerante a falhas, e balanceamento de carga.

Deste modo, criamos uma infraestrutura com um cluster web onde estarão duas máquinas proxys em modo ativo/passivo para realizar o balanceamento de carga das duas máquinas web onde encontram-se alojados os nossos sites de negócio. Um destes sites da empresa precisa de acesso a uma base de dados sql, sendo assim, criamos um cluster onde residem 2 máquinas sql em modo ativo/passivo. Posto isto, de forma a garantir tolerância a falhas temos sempre duas máquinas para o mesmo serviço, assim, se a que estiver ativa no momento tiver uma falha, não há falha na disponibilidade da infraestrutura. Também temos um sistema de raid nas máquinas web com 2 discos cada que em conjunto com o serviço glusterfs fazem a replicação dos dados armazenados tanto das máquinas web com as sql. Desta forma, garantimos o que é pretendido no paragrafo anterior.

A criação desta infraestrutura vai estar dividida por tópicos, e em cada um será abordado a forma de como foi realizada toda a instalação e configuração da mesma.

Cenário



Máquina Proxy

Instalação dos serviços:

Começamos por instalar todos os serviços que precisamos:

- Pacemaker
- Corosync
- HAProxy
- PCS

Configuração HAproxy

Para além das configurações default que se encontram no ficheiro `/etc/haproxy/haproxy.conf`, devemos acrescentar as seguintes configurações: (bind: <ip do cluster>:80)

```
frontend virtualip
    bind cluster:80
    mode http
    use_backend webservers

backend webservers
    mode http
    balance roundrobin
    server web1 192.168.97.121:80 check weight 1
    server web2 192.168.97.122:80 check weight 1
```

Configuracao Web Cluster

Autenticação dos nós:

Passamos para a autenticação dos nós, primeiramente vamos editar o ficheiro `/etc/hosts` e `/etc/hostname` em ambas as máquinas para ser mais fácil a nossa configuração (não ter de escrever ips várias vezes):

PROXY1:

```
GNU nano 4.8
127.0.0.1    localhost
192.168.97.100 proxy1
192.168.97.101 proxy2
# The following lines are desirable
```

```
GNU nano 4.8 /etc/hostname
proxy1
```

PROXY2:

```
GNU nano 4.8
127.0.0.1    localhost
192.168.99.101 proxy2
192.168.97.100 proxy1
```

```
GNU nano 4.8 /etc/hostname
proxy2
```

De seguida alterar a password do utilizador hacluster em ambas as máquinas com o comando `passwd`, para poder fazer a autenticação cluster com o comando `pcs host auth <nó1> <nó2>`:

```
root@proxy1:/home/osboxes# passwd hacluster
New password:
Retype new password:
passwd: password updated successfully
```

```
root@proxy1:/home/osboxes# pcs host auth proxy1 proxy2
Username: hacluster
Password:
proxy1: Authorized
proxy2: Authorized
```

Criação e inicialização do cluster:

Estando os nós autenticados podemos criar e inicializar o cluster com o seguinte comando (force é para forçar se existir algum erro):

`pcs cluster setup web-cluster proxy1 proxy2 --force`

`pcs cluster enable --all`

`pcs cluster start --all`

Propriedades do cluster:

Em todos os guias que encontrei na internet existem 2 propriedades que aparecem sempre, sendo assim acabei por optar em usar essas propriedades.

A primeira é a STONITH, o que é STONITH?

É uma propriedade dos clusters para manter a integridade dos nós em um cluster de alta disponibilidade, desliga automaticamente um nó que não está funcionando corretamente, sendo assim como estamos num ambiente de testes podemos desativar esta propriedade.

A segunda é a quórum policy é ignorada pois em um cluster de 2 nós, se a política quórum estiver habilitada e um dos nós falhar, o nó restante não poderá estabelecer a maioria dos votos de quórum necessários para executar serviços, portanto, não poderá assumir quaisquer recursos.

```
root@proxy1:/home/osboxes# pcs property set stonith-enabled=false
root@proxy1:/home/osboxes# pcs property set no-quorum-policy=ignore
root@proxy1:/home/osboxes# pcs property list
Cluster Properties:
cluster-infrastructure: corosync
cluster-name: web-cluster
dc-version: 2.0.3-4b1f869f0f
have-watchdog: false
no-quorum-policy: ignore
stonith-enabled: false
```

Criação, adição e configuração dos recursos:

Por fim temos de configurar dois recursos, o VirtualIP e HAProxy, ambos usando o provedor heartbeat para verificação do estado de funcionamento.

VirtualIP

O primeiro recurso, VirtualIP, permite-nos atribuir um endereço IP, neste caso IPv4, de forma virtual, que nos centraliza o ponto de entrada do tráfego.

pcs resource create VirtualIp ocf:heartbeat:IPaddr2 ip= cidr netmask=24 op monitor interval=30s

HAProxy

O segundo recurso, haproxy, permite-nos atribuir um serviço haproxy, com o binpath no diretório `/user/sbin/haproxy` ficheiro de configuração `/etc/haproxy/haproxy.conf`.

pcs resource create haproxy ocf:heartbeat:haproxy binpath=/user/sbin/haproxy conf=/etc/haproxy/haproxy.conf op monitor interval=15s

Configurações adicionais:

Adicionamos ambos os recursos a um grupo:

pcs resource group add ProxyGroupVirtualIp haproxy

Definimos a ordem de recuperação dos recursos, sendo que é prioritário a resolução de erro em primeiro lugar no virtual ip e sim depois no haproxy

pcs constraint order VirtualIp then haproxy

Este comando define que os dois recursos devem sempre correr no mesmo node:

pcs constraint colocation add VirtualIp with haproxy score=INFINITY

Status do cluster

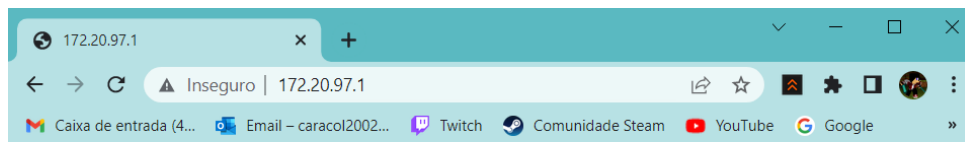
```
Cluster name: clusterWeb
Cluster Summary:
* Stack: corosync
* Current DC: proxy1 (version 2.0.3-4b1f869f0f) - partition with quorum
* Last updated: Fri Nov 25 10:55:10 2022
* Last change: Fri Nov 25 10:37:02 2022 by root via cibadmin on proxy1
* 2 nodes configured
* 2 resource instances configured

Node List:
* Online: [ proxy1 proxy2 ]

Full List of Resources:
* Resource Group: ProxyGroupVirtualIp:
  * haproxy (ocf::heartbeat:haproxy): Started proxy1
  * VirtualIp (ocf::heartbeat:IPaddr2): Started proxy1

Daemon Status:
corosync: active/enabled
pacemaker: active/enabled
pcsd: active/enabled
```

Prova de que o cliente tem acesso a página web do nginx a partir do ip virtual:



sistemas criticos

Máquinas Web

Criação e Configuração do RAID 1

Começamos por formatar os discos, para isso usou-se o comando `lsblk` para listar os discos disponíveis no sistema, de seguida formatamos os discos com sistema de ficheiro ext4 com o comando `mkfs.ext4`.

```
root@web1:/home/osboxes# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
loop0       7:0      0    4K  1 loop /snap/bare/5
loop1       7:1      0 248.8M  1 loop /snap/gnome-3-38-2004/99
loop2       7:2      0  91.7M  1 loop /snap/gtk-common-themes/1535
loop3       7:3      0  61.9M  1 loop /snap/core20/1328
loop4       7:4      0  65.2M  1 loop /snap/gtk-common-themes/1519
loop5       7:5      0   48M   1 loop /snap/snapd/17336
loop6       7:6      0 346.3M  1 loop /snap/gnome-3-38-2004/119
loop7       7:7      0  43.6M  1 loop /snap/snapd/14978
loop8       7:8      0  45.9M  1 loop /snap/snap-store/599
loop9       7:9      0  63.2M  1 loop /snap/core20/1695
loop10      7:10     0  54.2M  1 loop /snap/snap-store/558
sda         8:0      0 500G   0 disk
├─sda1      8:1      0 220.6G  0 part /
├─sda2      8:2      0  286M   0 part /boot
├─sda3      8:3      0    9G   0 part [SWAP]
└─sda4      8:4      0 270.1G  0 part /home
sdb         8:16     0    3G   0 disk
sdc         8:32     0    3G   0 disk
sr0        11:0     1 1024M   0 rom
```

```
root@web1:/home/osboxes# mkfs.ext4 /dev/sdb
mke2fs 1.45.5 (07-Jan-2020)
Creating filesystem with 786432 4k blocks and 196608 inodes
Filesystem UUID: b2f4c217-2478-461a-8f97-71bb54660642
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

root@web1:/home/osboxes# mkfs.ext4 /dev/sdc
mke2fs 1.45.5 (07-Jan-2020)
Creating filesystem with 786432 4k blocks and 196608 inodes
Filesystem UUID: 44404d57-7ac7-4904-b441-1ac8f6607a78
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done
```

Seguidamente a criação do RAID, para tal usamos o comando:

```
mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/sd{b,c}
```

Ou seja, criar um raid no bloco `/dev/md0`, de nível 1, com 2 discos sendo estes, `/dev/sdb` e `/dev/sbc`.

```
root@web1:/home/osboxes# mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/sd{b,c}
mdadm: /dev/sdb appears to contain an ext2fs file system
        size=3145728K  mtime=Wed Dec 31 19:00:00 1969
mdadm: Note: this array has metadata at the start and
        may not be suitable as a boot device.  If you plan to
        store '/boot' on this device please ensure that
        your boot-loader understands md/v1.x metadata, or use
        --metadata=0.90
mdadm: /dev/sdc appears to contain an ext2fs file system
        size=3145728K  mtime=Wed Dec 31 19:00:00 1969
Continue creating array? y
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
```

Agora formatamos o bloco com o sistema de ficheiros ext4 novamente, criamos a diretoria e fazemos a montagem do RAID na diretoria, com os seguintes comandos:

```
mkfs.ext4 /dev/md0
```

```
mkdir -p /raid1/
```

```
mount -t ext4 /dev/md0 /raid1
```

```

root@web1:/home/osboxes# mkfs.ext4 /dev/md0
mke2fs 1.45.5 (07-Jan-2020)
Creating filesystem with 785664 4k blocks and 196608 inodes
Filesystem UUID: b4e67f89-67ce-484a-831c-eb13e7779d9c
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

root@web1:/home/osboxes# mkdir -p /raid1
root@web1:/home/osboxes# mount -t ext4 /dev/md0 /raid1

```

Para confirmar se ficou tudo bem feito usamos novamente o comando: lsblk

```

sdb      8:16    0      3G  0 disk
└─md0    9:0      0      3G  0 raid1 /raid1
sdc      8:32    0      3G  0 disk
└─md0    9:0      0      3G  0 raid1 /raid1

```

Para existir total persistência de configuração no sistema, é adicionado o ponto de montagem no ficheiro `/etc/fstab`, desta forma, ao realizar um boot do sistema operativo, o ponto de montagem ir estar disponível automaticamente.

```

GNU nano 4.8 /etc/fstab
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name de
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options>        <dump> <pass>
# / was on /dev/sda1 during installation
UUID=2745355e-daeb-491a-8a15-06d08d03d8ba /      ext4    err
# /boot was on /dev/sda2 during installation
UUID=18a4e6bd-bec5-450a-9f01-8ba977aa7c5f /boot  ext4    def
# /home was on /dev/sda4 during installation
UUID=7b7b25b7-dcb6-41ee-8620-40e78673c1eb /home  ext4    def
# swap was on /dev/sda3 during installation
UUID=891c58c8-e4e3-443f-a9c1-422bd8fe7f12 none    swap    sw
# raid
UUID=3bfb9533-a790-40c9-880f-1eefb23ea0d8 /raid1 ext4    defaults 0 2

```

Para obter o UUID é só usar o comando: blkid /dev/md0

```

root@web1:/home/osboxes# blkid /dev/md0
/dev/md0: UUID="633d9205-4511-4e3b-842f-c6c4209fef17" TYPE="ext4"

```

Mudar o ficheiro de configuração do mdadm.conf:

```
GNU nano 4.8 /etc/mdadm/mdadm.conf
# mdadm.conf
#
# !NB! Run update-initramfs -u after updating this file.
# !NB! This will ensure that initramfs has an uptodate copy.
#
# Please refer to mdadm.conf(5) for information about this file.
#
# by default (built-in), scan all partitions (/proc/partitions) and all
# containers for MD superblocks. alternatively, specify devices to scan, using
# wildcards if desired.
#DEVICE partitions containers
CREATE owner=root group=disk mode=0660 auto=yes
# automatically tag new arrays as belonging to the local system
HOMEHOST <system>
# instruct the monitoring daemon where to send mail alerts
MAILADDR root
# definitions of existing MD arrays
# This configuration was auto-generated on Tue, 15 Nov 2022 06:51:28 -0500 by mkconf
ARRAY /dev/md0 metadata=1.2 name=web2:0 UUID=a4e41d52:429cca06:18d83068:f5be6463
```

Para adicionar a última linha do ficheiro é só usar o seguinte comando:

```
root@web2:/home/osboxes# mdadm --detail --scan >> /etc/mdadm/mdadm.conf
```

Ao dar reboot à máquina o md0 passa a ser md127, para evitar essa mudança temos de usar o comando:

```
root@web2:/home/osboxes# update-initramfs -u -k all
```

Verificação da instalação RAID1 na web1:

```
sda      8:0    0   500G   0 disk
├─sda1   8:1    0  220.6G   0 part  /
├─sda2   8:2    0   286M   0 part  /boot
├─sda3   8:3    0    9G    0 part  [SWAP]
└─sda4   8:4    0  270.1G   0 part  /home
sdb      8:16   0    3G    0 disk
└─md0    9:0    0    3G    0 raid1  /raid1
sdc      8:32   0    3G    0 disk
└─md0    9:0    0    3G    0 raid1  /raid1
sr0     11:0    1  1024M   0 rom
root@web1:/home/osboxes#
```

Verificação da instalação RAID1 na web2:

```
sda      8:0    0   500G   0 disk
├─sda1   8:1    0  220.6G   0 part  /
├─sda2   8:2    0   286M   0 part  /boot
├─sda3   8:3    0    9G    0 part  [SWAP]
└─sda4   8:4    0  270.1G   0 part  /home
sdb      8:16   0    3G    0 disk
└─md0    9:0    0    3G    0 raid1  /raid1
sdc      8:32   0    3G    0 disk
└─md0    9:0    0    3G    0 raid1  /raid1
sr0     11:0    1  1024M   0 rom
root@web2:/home/osboxes#
```

Configuração Nginx:

```
GNU nano 4.8                                     default
# Default server configuration
#
server {
    listen 80 default_server;
    listen [::]:80 default_server;

    # SSL configuration
    #
    # listen 443 ssl default_server;
    # listen [::]:443 ssl default_server;
    #
    # Note: You should disable gzip for SSL traffic.
    # See: https://bugs.debian.org/773332
    #
    # Read up on ssl_ciphers to ensure a secure configuration.
    # See: https://bugs.debian.org/765782
    #
    # Self signed certs generated by the ssl-cert package
    # Don't use them in a production server!
    #
    # include snippets/snakeoil.conf;

    root /raid1/cluster/www/;

    # Add index.php to the list if you are using PHP
```

O root do website será a diretoria que estará sobre o RAID e consequentemente sobre um volume glusterfs, de forma a garantir redundância entres servidores.

Configuracao GlusterFS

Instalação:

Comando para instalar o gluster server:

```
root@web1:/home/osboxes# apt install glusterfs-server -y
Reading package lists... Done
Building dependency tree
```

Comandos para iniciar o serviço e colocá-lo a iniciar sempre no arranque na máquina, respetivamente:

```
root@web1:/home/osboxes# systemctl start glusterd.service
root@web1:/home/osboxes# systemctl enable glusterd.service
Created symlink /etc/systemd/system/multi-user.target.wants/glusterd.service → /lib/systemd/system/glusterd.service.
```

Autenticação das máquinas:

```
root@web1:/home/osboxes# gluster peer probe web2
peer probe: success.
```

```
root@web2:/home/osboxes# gluster peer probe web1
peer probe: success. Host web1 port 24007 already in peer list
```

Criação do volume e configuração:

```
root@web1:/home/osboxes# gluster volume create webvolume replica 2 web1:/raid1/cluster/www/
web2:/raid1/cluster/www/ force
volume create: webvolume: success: please start the volume to access data
```

Arrancar o volume e os status:

```
root@web1:/home/osboxes# gluster volume start webvolume
volume start: webvolume: success
```

```
root@web1:/home/osboxes# gluster volume status webvolume
Status of volume: webvolume
Gluster process                TCP Port  RDMA Port  Online  Pid
-----
Brick web1:/raid1/cluster/www  49152     0           Y       17777
Brick web2:/raid1/cluster/www  49152     0           Y       97913
Self-heal Daemon on localhost  N/A       N/A         Y       17798
Self-heal Daemon on web2      N/A       N/A         Y       97934

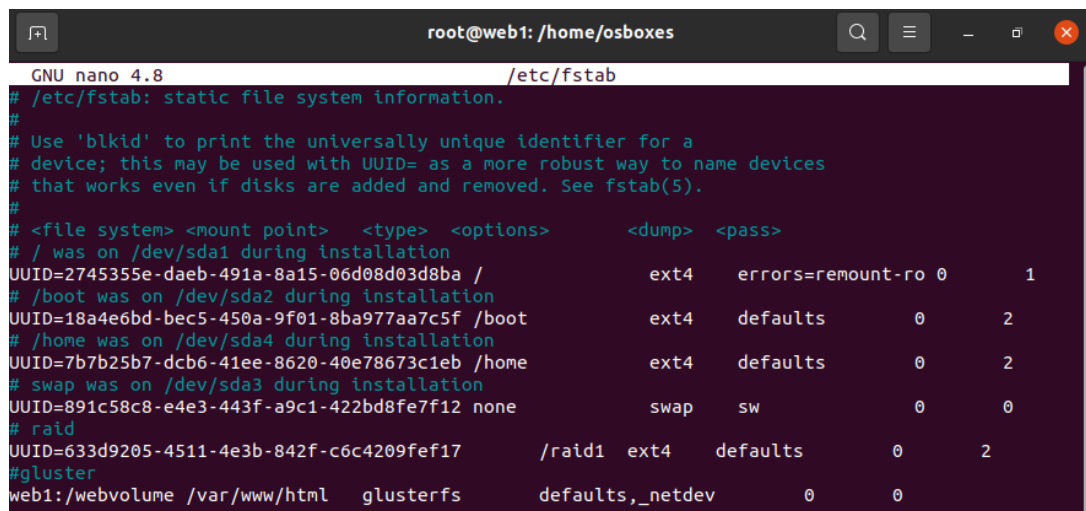
Task Status of Volume webvolume
-----
There are no active volume tasks
```

Montagem do diretório comum a ambos os nós que contem os ficheiros do website, neste caso o nginx:

```
root@web1:~# mount -t glusterfs web1:/webvolume /var/www/html
```

```
root@web2:/var/www/html# mount -t glusterfs web2:/webvolume /var/www/html
```

Mudar ficheiro fstab outra vez



```
root@web1:/home/osboxes
GNU nano 4.8 /etc/fstab
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options>          <dump> <pass>
# / was on /dev/sda1 during installation
UUID=2745355e-daeb-491a-8a15-06d08d03d8ba /
# /boot was on /dev/sda2 during installation
UUID=18a4e6bd-bec5-450a-9f01-8ba977aa7c5f /boot
# /home was on /dev/sda4 during installation
UUID=7b7b25b7-dcb6-41ee-8620-40e78673c1eb /home
# swap was on /dev/sda3 during installation
UUID=891c58c8-e4e3-443f-a9c1-422bd8fe7f12 none
# raid
UUID=633d9205-4511-4e3b-842f-c6c4209fef17 /raid1 ext4 defaults 0 2
#gluster
web1:/webvolume /var/www/html glusterfs defaults,_netdev 0 0
```

Site da empresa

Ferramenta Cacti

Instalação:

Antes de instalar a ferramenta devemos instalar algumas dependências:

apt-get install snmp php-snmp rrdtool librrds-perl unzip curl git gnupg2 -y

Instalação apache2:

Instalar o web server apache2 service e inicializá-lo.

Apt-get install apache2

Systemctl start apache2

Systemctl enable apache2

Instalação PHP e Modules:

Para a instalação do php e alguns modelos associados adicionamos o repositório ppa:onrej/php

Add-apt-repository ppa:onrej/php

De seguida instalámos os mesmos.

apt-get install php php-mysql libapache2-mod-php php-xml php-ldap php-mbstring php-gd php-gmp -y

Alterar os seguintes ficheiros, /etc/php/7.4/apache2/php.ini | /etc/php/7.4/cli/php.ini, com os dados seguintes.

memory_limit = 512M

max_execution_time = 60

date.timezone = Asia/Kolkata

Configuração MariaDB:

A ferramenta cacti usa a mariadb com backend database, por isso temos de criar uma base de dados e um user para o cacti nas nossas máquinas sql.

Primeiro começamos por alterar o ficheiro de configuração de mariadb:

nano /etc/mysql/mariadb.conf.d/50-server.cnf

Adicionar e mudar a seguinte informação:

```
GNU nano 4.8 /etc/mysql/mariadb.conf.d/50-server.cnf
[mariadb]
# This group is only read by MariaDB-10.3 servers.
# If you use the same .cnf file for MariaDB of different versions,
# use this group for options that older servers don't understand
[mariadb-10.3]

collation-server = utf8mb4_unicode_ci
max_heap_table_size = 128M
tmp_table_size = 64M
join_buffer_size = 64M
innodb_file_format = Barracuda
innodb_large_prefix = 1
innodb_buffer_pool_size = 512M
innodb_flush_log_at_timeout = 3
innodb_read_io_threads = 32
innodb_write_io_threads = 16
innodb_io_capacity = 5000
innodb_io_capacity_max = 10000
```

Entrar na Shell do mariadb e criar a base de dados e o user:

mysql

```
MariaDB [(none)]> create database cactidb;
Query OK, 1 row affected (0.013 sec)

MariaDB [(none)]> GRANT ALL ON cactidb.* TO cactiuser@localhost IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.008 sec)

MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.007 sec)

MariaDB [(none)]> GRANT ALL ON cactidb.* TO cactiuser@web1 IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.004 sec)

MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.006 sec)

MariaDB [(none)]> exit;
Bye
```

Importar timezone data para o mysql database:

mysql mysql < /usr/share/mysql/mysql_test_data_timezone.sql

Conceder privilégios necessários no timezone do mysql:

mysql

```
MariaDB [(none)]> GRANT SELECT ON mysql.time_zone_name TO cactiuser@localhost;  
Query OK, 0 rows affected (0.002 sec)  
  
MariaDB [(none)]> GRANT SELECT ON mysql.time_zone_name TO cactiuser@web1;  
Query OK, 0 rows affected (0.002 sec)  
  
MariaDB [(none)]> flush privileges;  
Query OK, 0 rows affected (0.004 sec)  
  
MariaDB [(none)]> exit;  
Bye
```

Instalação e configuração do Cacti:

Máquina sql:

Aqui iremos importar a base de dados para a que criamos (cactidb):

```
root@sql1:/home/osboxes# wget https://www.cacti.net/downloads/cacti-latest.tar.gz  
z  
  
root@sql1:/home/osboxes# tar -zxvf cacti-latest.tar.gz  
cacti-1.2.22/  
  
root@sql1:/home/osboxes# mysql cactidb < /home/osboxes/cacti-1.2.22/cacti.sql
```

Máquina web:

Fazer download da última versão do cacti.

```
root@web1:/home/osboxes# wget https://www.cacti.net/downloads/cacti-latest.tar.gz  
2022-12-04 05:36:44 https://www.cacti.net/downloads/cacti-latest.tar.gz
```

Extrair os ficheiros e movê-los para o diretório Apache root:

```
root@web1:/home/osboxes# tar -zxvf cacti-latest.tar.gz  
cacti-1.2.22/  
  
root@web1:/home/osboxes# mkdir /var/www/html/cacti  
root@web1:/home/osboxes# mv cacti-1.2.22/* /var/www/html/cacti
```

De seguida editar o ficheiro de config do cacti e mudar a informação la pedida para a informação se criou anteriormente, como por exemplo o nome da base de dados, etc:


```

$database_type      = 'mysql';
$database_default   = 'cactidb';
$database_hostname   = '192.168.97.111';
$database_username   = 'cactiuser';
$database_password   = 'password';
$database_port       = '3306';
$database_retries    = 5;
$database_ssl        = false;
$database_ssl_key     = '';
$database_ssl_cert    = '';
$database_ssl_ca      = '';
$database_persist     = false;

/*
 * When the cacti server is a remote poller
 * the main cacti server. Otherwise, these
 * must remain commented out.
 */

$rdatabase_type      = 'mysql';
$rdatabase_default   = 'cactidb';
$rdatabase_hostname   = '192.168.97.111';
$rdatabase_username   = 'cactiuser';
$rdatabase_password   = 'password';
$rdatabase_port       = '3306';
$rdatabase_retries    = 5;
$rdatabase_ssl        = false;
$rdatabase_ssl_key     = '';
$rdatabase_ssl_cert    = '';
$rdatabase_ssl_ca      = '';

```

Alterar as permissões do diretório do cacti:

```

root@web1:/var/www/html/cacti# chown -R www-data:www-data /var/www/html/cacti/
root@web1:/var/www/html/cacti# chmod -R 775 /var/www/html/cacti/

```

Criar um ficheiro cron job para o cacti:

*** /5 * * * * www-data php /var/www/html/cacti/poller.php > /dev/null 2>&1**

Configuração Apache para Cacti:

Criar o ficheiro Apache virtual host conf para o Cacti:

nano /etc/apache2/sites-available/cacti.conf

```

GNU nano 4.8
Alias /cacti /var/www/html/cacti

<Directory /var/www/html/cacti>
    Options +FollowSymLinks
    AllowOverride None
    <IfVersion >= 2.3>
        Require all granted
    </IfVersion>
    <IfVersion < 2.3>
        Order Allow,Deny
        Allow from all
    </IfVersion>

    AddType application/x-httpd-php .php

<IfModule mod_php.c>
    php_flag magic_quotes_gpc Off
    php_flag short_open_tag On
    php_flag register_globals Off
    php_flag register_argc_argv On
    php_flag track_vars On
    # this setting is necessary for some locales
    php_value mbstring.func_overload 0
    php_value include_path .
</IfModule>

    DirectoryIndex index.php
</Directory>

```

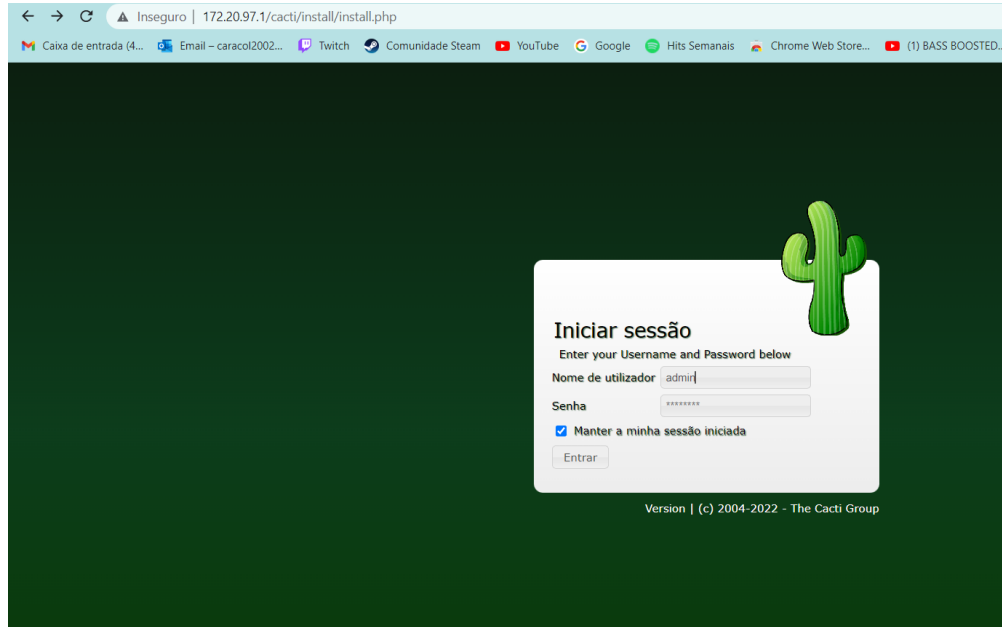
Guardar e ligar o virtual host:

a2ensite cacti

Reniciar o apache:

Systemctl restart apache2

Prova de funcionamento:



Ferramenta Ganglia:

Instalação:

Para começar fazemos update ao sistema.

`apt-get update`

Instalação apache2:

Este passo já foi feito na ferramenta anterior.

Instalação PHP e Modules:

Apesar de já termos o PHP e alguns modelos instalados, voltei a instalar de novo alguns para confirmar se ficava tudo bem instalado.

De seguida instalámos os mesmos.

`apt-get install php7.2 libapache2-mod-php7.2 php7.2-common php7.2-gmp php7.2-curl`

`apt-get install php7.2-intl php7.2-mbstring php7.2-xmlrpc php7.2-mysql php7.2-gd`

`apt-get install php7.2-xml php7.2-cli php7.2-zip`

Configuração do ficheiro php default do apache:

Verificar e mudar os seguintes valores.

Nano /etc/php/7.2/apache2/php.ini

file_uploads = On

allow_url_fopen = On

short_open_tag = On

memory_limit = 256M

upload_max_filesize = 100M

max_execution_time = 360

max_input_vars = 1500

date.timezone = America/New_York

Reniciar o apache2

systemctl restart apache2.service

Instalação do ganglia:

Instalar os recursos:

apt-get install ganglia-monitor rrdtool gmetad ganglia-webfrontend

Iniciar e dar enable ao serviço:

systemctl start ganglia-monitor.service

systemctl enable ganglia-monitor.service

Configuração do ficheiro gmond.conf no ganglia server:

```
cluster {
  name = "my cluster"
  owner = "unspecified"
  latlong = "unspecified"
  url = "unspecified"
}

/* used to only support having a single channel */
udp_send_channel {
  /* mcast_join = 239.2.11.71 */
  port = 8649
  ttl = 1
}

/* You can specify as many udp_recv_channels as you like as well. */
udp_recv_channel {
  /* mcast_join = 239.2.11.71 */
  port = 8649
  /* bind = 239.2.11.71 */
}

/* You can specify as many tcp_accept_channels as you like to share
   an xml description of the state of the cluster */
tcp_accept_channel {
  port = 8649
}
```

Configuração do conf file do ganglia no apache2:

cp /etc/ganglia-webfrontend/apache.conf /etc/apache2/sites-enabled/ganglia.conf

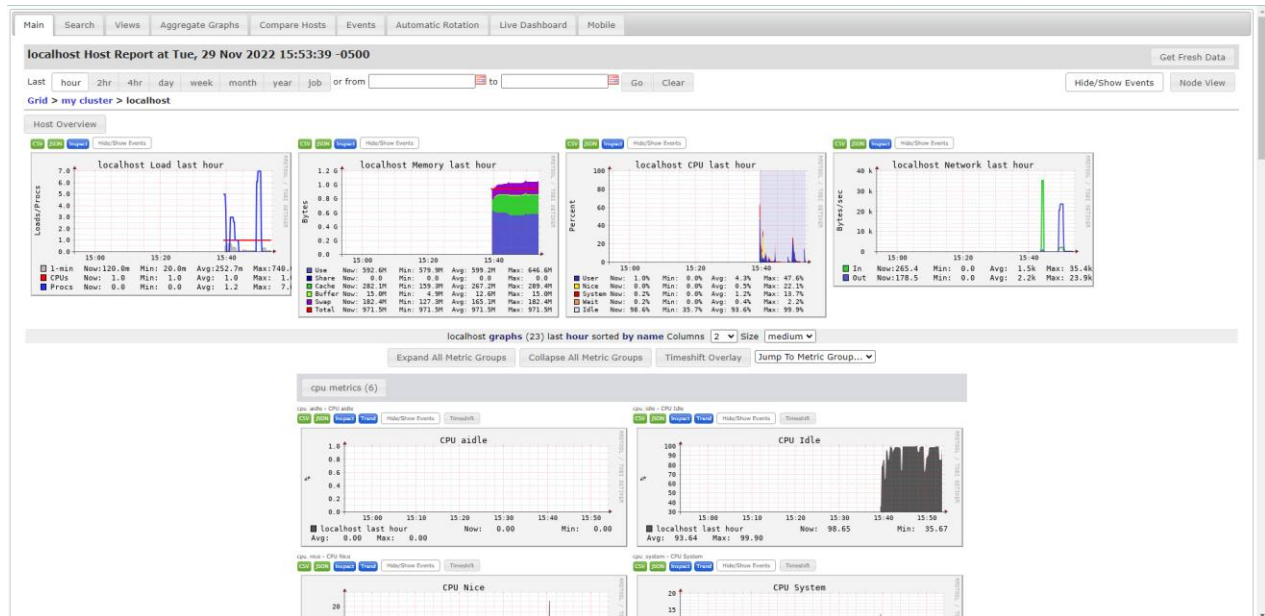
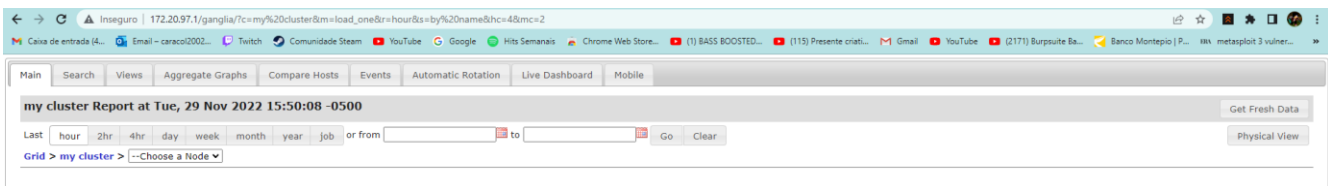
Reiniciar os serviços:

systemctl restart ganglia-monitor

systemctl restart gmetad

systemctl restart apache2

Prova de funcionamento:



Instalar ganglia nos clientes:

Neste caso os clientes serão o resto das nossas máquinas.

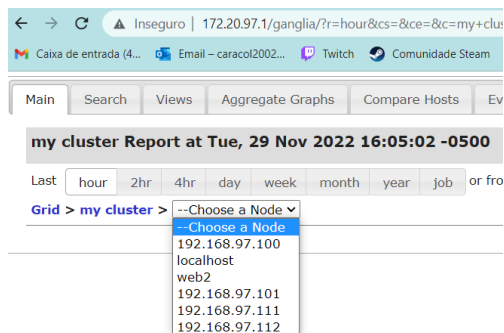
apt-get install ganglia-monitor

nano /etc/ganglia/gmond.conf

```
/* Feel free to specify as many udp_send_channels as you like. Gmond
used to only support having a single channel */
udp_send_channel {
    host = 192.168.97.121
    port = 8649
    ttl = 1
}
```

systemctl start ganglia-monitor

Assim feito esta instalação devem aparecer todas as máquinas:



Configuração máquinas SQL

Instalação MariaDB

Para instalar o serviço usamos o comando seguinte:

```
root@sql1:~# apt install mariadb-server mariadb-client -y
```

Comando enable para que sempre que a máquina seja reiniciada ou ligada o serviço inicie automaticamente.

Comando start para que o serviço seja iniciado.

Comando status para que mostre se o serviço está ligado, desligado ou com algum erro.

```
root@sql1:~# systemctl enable mariadb
root@sql1:~# systemctl start mariadb
root@sql1:~# systemctl status mariadb
● mariadb.service - MariaDB 10.3.34 database server
   Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor pres>
   Active: active (running) since Sat 2022-11-19 15:19:14 EST; 7min ago
     Docs: man:mysqld(8)
           https://mariadb.com/kb/en/library/systemd/
   Main PID: 3167 (mysqld)
    Status: "Taking your SQL requests now..."
     Tasks: 30 (limit: 1086)
    Memory: 69.6M
    CGroup: /system.slice/mariadb.service
            └─3167 /usr/sbin/mysqld

Nov 19 15:19:14 sql1 mysqld[3167]: 2022-11-19 15:19:14 0 [Note] /usr/sbin/mysq>
Nov 19 15:19:14 sql1 systemd[1]: Started MariaDB 10.3.34 database server.
Nov 19 15:19:14 sql1 /etc/mysql/debian-start[3202]: Upgrading MySQL tables if >
Nov 19 15:19:14 sql1 /etc/mysql/debian-start[3205]: Looking for 'mysql' as: /u>
Nov 19 15:19:14 sql1 /etc/mysql/debian-start[3205]: Looking for 'mysqlcheck' a>
Nov 19 15:19:14 sql1 /etc/mysql/debian-start[3205]: This installation of Maria>
Nov 19 15:19:14 sql1 /etc/mysql/debian-start[3205]: There is no need to run my>
Nov 19 15:19:14 sql1 /etc/mysql/debian-start[3205]: You can use --force if you>
Nov 19 15:19:14 sql1 /etc/mysql/debian-start[3213]: Checking for insecure root>
Nov 19 15:19:14 sql1 /etc/mysql/debian-start[3219]: Triggering myisam-recover >
lines 1-22/22 (END)
```

Configuração MariaDB

Configuração do é realizada a instalação segura onde se indica alguns parâmetros de segurança:

Escrevemos “Y” para poder colocar de seguida as credencias para o root da base de dados.

```
root@sql1:~# mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MariaDB
root user without the proper authorisation.

You already have a root password set, so you can safely answer 'n'.

Change the root password? [Y/n] y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
... Success!
```

Escrevemos “y” em todos os parâmetros para com que:

- 1- Remover anonymous users;
- 2- Desativar login remoto;
- 3- Remover a base de dados de testes que vem com a instalação por default;
- 4- Recarregar as tabelas de privilégios para que as alterações entrem vigor imediatamente;

```
Remove anonymous users? [Y/n] y
... Success!

Normally, root should only be allowed to connect from 'localhost'. This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] y
... Success!

By default, MariaDB comes with a database named 'test' that anyone can
access. This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] y
- Dropping test database...
... Success!
- Removing privileges on test database...
... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] y
... Success!

Cleaning up...
```

Para finalizar configuramos um usuário administrativo com autenticação por password e atribuímos todos os privilégios administrativos a esse usuário.

Para tal fazemos os seguintes comandos:

- 1- Login como root;
- 2- Criamos um user chamamos 'admin_user' com uma password associada;

```
root@sql1:~# mariadb -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 46
Server version: 10.3.34-MariaDB-0ubuntu0.20.04.1 Ubuntu 20.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE USER 'admin_user'@'localhost' IDENTIFIED BY 'XXXXXXXXXX';
Query OK, 0 rows affected (0.000 sec)
```

- 3- Atribuímos todos os privilégios root da base de dados aos user; (o wild-card '*.*' indica que o user tem permissão para executar qualquer tarefa em qualquer base de dados e no servidor respetivo);

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON *.* TO 'admin_user'@'localhost';
Query OK, 0 rows affected (0.000 sec)
```

- 4- Para aplicar as alterações;

```
MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.000 sec)
```

Validação da criação do user e bom funcionamento do serviço:

```
root@sql1:~# mariadb -u admin_user -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 49
Server version: 10.3.34-MariaDB-0ubuntu0.20.04.1 Ubuntu 20.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
+-----+
3 rows in set (0.011 sec)
```

```
MariaDB [(none)]> SELECT host, user FROM mysql.user;
+-----+-----+
| host      | user      |
+-----+-----+
| localhost | admin_user |
| localhost | root      |
+-----+-----+
2 rows in set (0.000 sec)
```

```
MariaDB [(none)]> CREATE DATABASE test_db;
Query OK, 1 row affected (0.000 sec)
```

Configuração SQL Cluster

Instalação:

Começamos por instalar todos os serviços que precisamos:

- Pacemaker
- Corosync
- PCS

Autenticação dos nós:

Alterar o ficheiro /etc/hosts:

Sql1:

```
GNU nano 4.8
127.0.0.1    localhost
192.168.97.111 sql1
192.168.97.112 sql2
```

Sql2:

```
GNU nano 4.8
127.0.0.1    localhost
192.168.97.112 sql2
192.168.97.111 sql1
```

Alterar a password do user “hacluster” em ambos: **passwd hacluster**

Para autenticar os nós usamos o comando:

pcs host auth sql1 sql2

Criação do cluster:

Para a criação do cluster denominado de “clusterSql” em ambos os nós sql1 e sql2 usamos o comando:

pcs cluster setup clusterSql sql1 sql2 --force

Inicialização do cluster:

pcs cluster enable --all

pcs cluster start --all

Propriedades do cluster:

Pelos motivos previamente explicados na configuração das máquinas proxy, são desabilitadas 2 propriedades com os seguintes comandos:

pcs property set stonith-enabled=false

pcs property set no-quorum-policy=ignore

Criação, adição e configuração de recursos:

Endereço IP virtual:

Usamos o mesmo comando usado na configuração do cluster web, mas só mudamos o IP:

pcs resource create VirtualIP ocf:heartbeat:IPaddr2 ip=192.168.49.110 cidr netmask=24 op monitor interval=5s

pcs resource enable VirtualIP

MariaDB:

Para este recurso usamos o provedor systemd invés do heartbeat devido há natureza do recurso ser sobre um serviço, a configuração é ligeiramente mais facilitada devido a só termos de indicar os tempos de monitorização, este recurso é também adicionado ao grupo “maria-group”. Para tal usamos o seguinte comando:

pcs resource create mariadb systemd:mariadb op start timeout=60s op stop timeout=60s op monitor interval=20s timeout=30s --group mariadb-group

De seguida indicamos a ordem de inicialização e a sua “colocação”, ou seja, ambos os recursos vão correr na mesma máquina. Comandos:

pcs constraint order start VirtualIP then mariadb-group

pcs constraint colocation add VirtualIP with mariadb-server score=INFINITY

Status do cluster:

```
root@sql1:/home/osboxes# pcs status
Cluster name: sqlCluster
Cluster Summary:
 * Stack: corosync
 * Current DC: sql1 (version 2.0.3-4b1f869f0f) - partition with quorum
 * Last updated: Tue Nov 22 14:17:03 2022
 * Last change: Tue Nov 22 13:49:28 2022 by root via cibadmin on sql1
 * 2 nodes configured
 * 2 resource instances configured

Node List:
 * Online: [ sql1 sql2 ]

Full List of Resources:
 * VirtualIP (ocf::heartbeat:IPaddr2): Started sql1
 * Resource Group: mariadb-group:
   * mariadb (systemd:mariadb): Started sql1

Daemon Status:
 corosync: active/enabled
 pacemaker: active/enabled
 pcsd: active/enabled
```

Configuração do gluster

Neste caso as nossas máquinas SQL vão ser clientes do nosso serviço master gluster, ou seja, web1.

Criação e inicialização do volume:

Sendo assim teremos de voltar à máquina web1 para realizar a criação do volume na máquina web1.

gluster volume create sqlstorage replica 2 sql1:/raid1/cluster/sql sql2:/raid1/cluster/sql

gluster volume start sqlstorage

Montagem de volumes:

Esta montagem é respetiva os nossos clientes gluster, ou seja, as máquinas SQL, garantindo assim a replicação dos conteúdos entre os servidores.

mount -t glusterfs web1:/sqlstorage /var/lib/mysql -o nodev,nofail

mount -t glusterfs web2:/sqlstorage /var/lib/mysql -o nodev,nofail

É também importante as montagens serem indicadas no ficheiro /etc/fstab, para continuar a garantir a replicação de dados.

web1:/sqlstorage /var/lib/mysql glusterfs defaults, nodev,nofail 0 0 (noatime)

web2:/sqlstorage /var/lib/mysql glusterfs defaults, nodev,nofail 0 0 (noatime)

Ferramenta para efetuar x números de pedidos por segundo

Apache JMeter

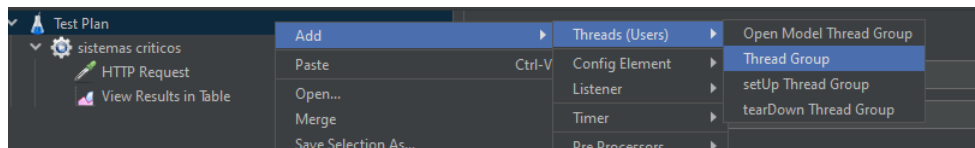
Instalação:

Fazer download do site oficial: https://jmeter.apache.org/download_jmeter.cgi

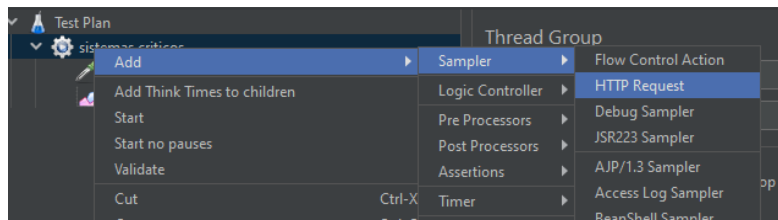
Extrair os ficheiros e no meu caso como estou no Windows 10 para correr a ferramenta simplesmente tenho de abrir o ficheiro .bat que se encontra na pasta /bin

Configuração

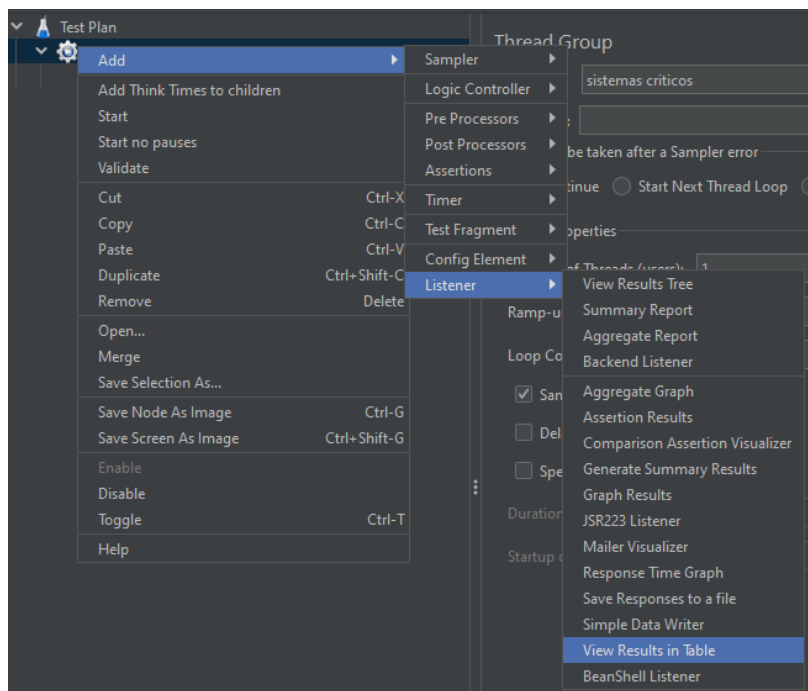
Criamos a nossa thread group denominada por “sistemas criticos”:



Adicionamos a essa thread group um pedido http request:



E por fim um view para visualizar as nossas métricas:



O nosso pedido http vai ser direcionado ao site da empresa (ganglia):

The screenshot shows a web application for creating HTTP requests. At the top, there's a section for 'HTTP Request' with a 'Name' field containing 'HTTP Request' and an empty 'Comments' field. Below this are two tabs: 'Basic' (selected) and 'Advanced'. The 'Basic' tab contains a 'Web Server' section with 'Protocol [http]:' set to 'http' and 'Server Name or IP:' set to '172.20.97.1'. Underneath is an 'HTTP Request' section with a dropdown menu set to 'GET' and a 'Path:' field containing '/ganglia/'. Below the path field are several checkboxes: 'Redirect Automatically' (unchecked), 'Follow Redirects' (checked), 'Use KeepAlive' (checked), 'Use multipart/form-data' (unchecked), and 'Browser-compatible headers' (unchecked). At the bottom of the 'Basic' tab are three sub-tabs: 'Parameters' (selected), 'Body Data', and 'Files Upload'. The 'Parameters' sub-tab shows a table header with 'Name:', 'Value', and 'URL Encode:'. Above the table is a label 'Send Parameters With the Request:'.

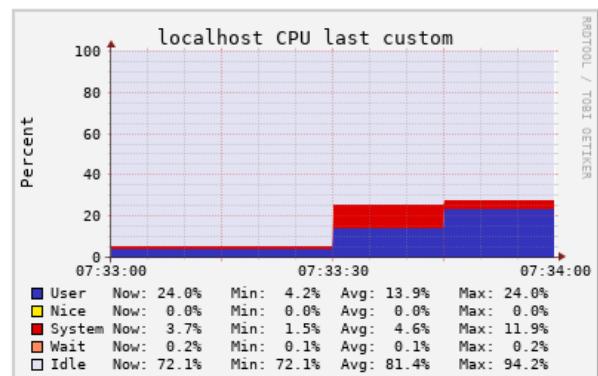
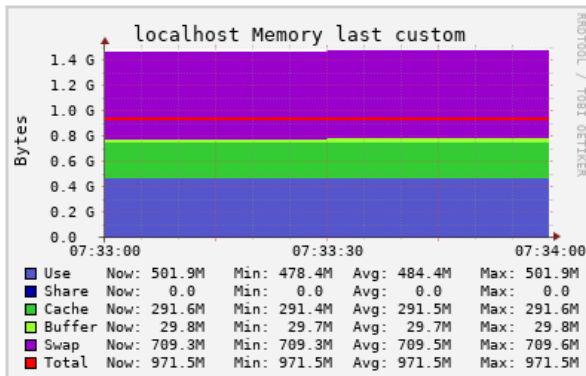
Name:	Value	URL Encode:
-------	-------	-------------

Avaliar o impacto que estes pedidos têm na memória RAM e no CPU

1 Servidor

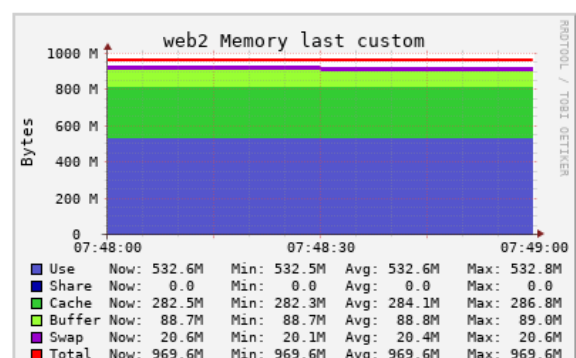
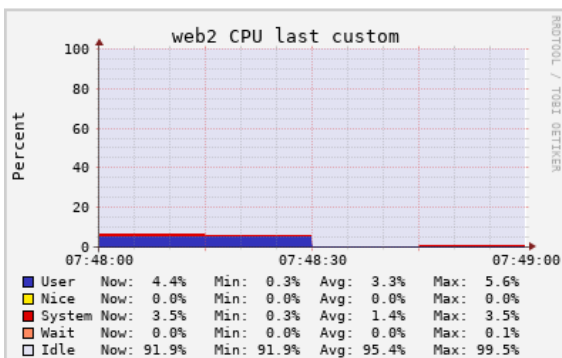
Imagens do comportamento do CPU e da RAM ao realizar os pedidos http request:

Podemos ver que a memoria RAM ficou estável, sem nenhuma oscilação brusca.



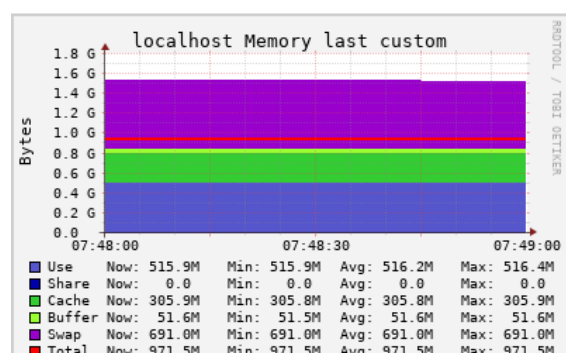
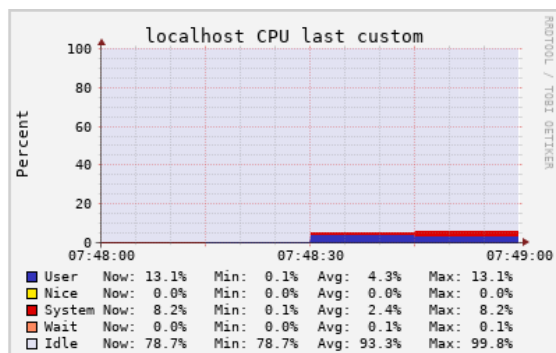
Ao contrário da RAM podemos ver que o CPU teve uma ligeira subida, a medida que se ia aumentando o número de pedido o uso do CPU também aumentava

2 Servidores



Em ambos os servidores a memoria RAM continua estável sem nenhuma oscilação brusca.

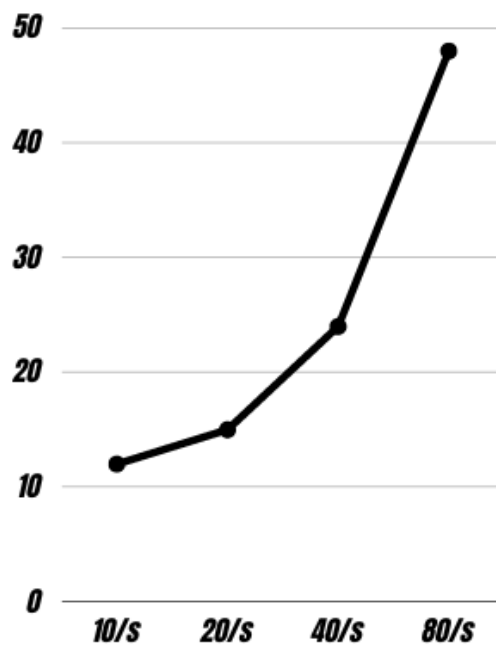
Agora como temos os 2 servidores em balanceamento de carga podemos ver que os pedidos foram divididos por ambos os servidores, os primeiros 30 segundos foram para web2 e os restantes para o web1, e não temos nenhuma oscilação brusca como temos com usamos só um servidor



Análise qualitativa desses resultados

Balanceamento com 2 servidores:

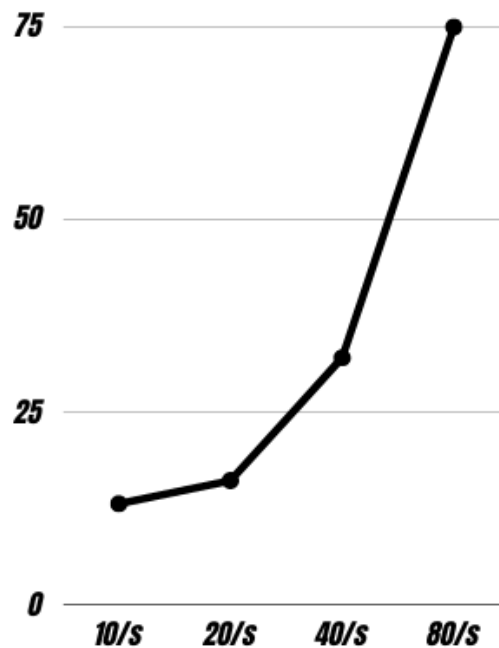
Até mesmo para o leitor que não tenha conhecimento na área talvez seja possível compreender o porquê do uso de dois servidores para balanceamento de carga, mas uma representação visual é o que realmente permite uma boa compreensão.



Com a representação deste gráfico, é possível ver um aumento não muito elevado e previsível consoante os pedidos enviados, tendo a latência como medida de capacidade de reposta. Sendo estes servidores “balanceados” os pedidos são divididos entre os dois para garantir que não haja uma sobrecarga num servidor, em relação aos custos à empresa, sim fica mais barato ter só um servidor, mas para ter só um servidor que não tenha sobrecarga tem de ser um com melhor hardware o que aumenta o seu preço também, e uma das grandes vantagens de ter 2 servidores é que conseguimos garantir a alta disponibilidade, balanceamento de carga e a tolerância a falhas.

Inexistência de balanceamento (1 servidor)

No caso onde não existe balanceamento, uma curva mais acentuada é facilmente visível, indicando que é mais difícil, se não impossível, prever com alguma qualidade a capacidade de resposta e até mesmo o ponto de rutura do servidor, visto que apenas existe um em funcionamento e este é sobrecarregado mais facilmente.



Não só a latência aumenta exponencialmente, o mesmo pode ocorrer com os custos, sendo necessária mais manutenção ou melhor hardware para garantir o funcionamento do servidor, mencionando ainda que a inexistência de outro servidor, causa um ponto único de falha, afetando a disponibilidade e segurança do negócio.

NOTA: Para a criação destes gráficos tive como apoio a análise das imagens que se encontram nos anexos.

Conclusão

Através do desenvolvimento deste trabalho prático foi possível aprofundar conhecimentos relativos aos conteúdos abordados nas aulas. Aprendi bastantes coisas sobre análise de recursos RAM, CPU etc, como configurar bastantes ferramentas, balanceadores de carga, clusters e replicação de ficheiros entre máquinas.

De uma forma geral acho que foram cumpridos todos os objetivos propostos no enunciado do trabalho prático.

Anexos

Imagens das métricas capturadas pelo Apache JMeter (2 servidores / 1 servidor – 10 pedidos por segundo)

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency ↓	Connect Time(ms)
1	14:48:10.324	Thread Group 1-1	HTTP Request ga...	102	✓	27958	121	65	5
2	14:48:10.426	Thread Group 1-1	HTTP Request ga...	94	✓	12345	121	38	0
3	14:48:10.520	Thread Group 1-1	HTTP Request ga...	40	✓	27958	121	16	0
5	14:48:10.617	Thread Group 1-1	HTTP Request ga...	39	✓	27958	121	15	0
9	14:48:10.816	Thread Group 1-1	HTTP Request ga...	42	✓	27951	121	15	0
8	14:48:10.759	Thread Group 1-1	HTTP Request ga...	57	✓	12345	121	14	0
4	14:48:10.561	Thread Group 1-1	HTTP Request ga...	56	✓	12345	121	13	0
7	14:48:10.715	Thread Group 1-1	HTTP Request ga...	44	✓	27958	121	13	0
10	14:48:10.858	Thread Group 1-1	HTTP Request ga...	60	✓	12345	121	13	0
6	14:48:10.657	Thread Group 1-1	HTTP Request ga...	58	✓	12345	121	12	0

View Results in Table

Name:

1 server 1 user 10 pedidos

Comments:

Write results to file / Read from file

Filename





















Browse...





















Log/Display Only: ☐ Errors ☐ Successes

Configure



























Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
1	14:13:35.146	Thread Group 1-1	HTTP Request ga...	90	✓	12345	121	32	1
2	14:13:35.236	Thread Group 1-1	HTTP Request ga...	64	✓	12345	121	15	0
3	14:13:35.300	Thread Group 1-1	HTTP Request ga...	61	✓	12345	121	14	0
4	14:13:35.361	Thread Group 1-1	HTTP Request ga...	54	✓	12345	121	14	0
5	14:13:35.416	Thread Group 1-1	HTTP Request ga...	50	✓	12345	121	12	0
6	14:13:35.467	Thread Group 1-1	HTTP Request ga...	54	✓	12345	121	12	0
7	14:13:35.521	Thread Group 1-1	HTTP Request ga...	56	✓	12345	121	13	0
8	14:13:35.577	Thread Group 1-1	HTTP Request ga...	63	✓	12345	121	15	0
9	14:13:35.640	Thread Group 1-1	HTTP Request ga...	60	✓	12345	121	14	0
10	14:13:35.700	Thread Group 1-1	HTTP Request ga...	54	✓	12345	121	13	0



























Imagens das métricas capturadas pelo Apache JMeter (2 servidores / 1 servidor – 20 pedidos por segundo)

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency ↓	Connect Time(ms)
1	14:51:00.043	Thread Group 1-1	HTTP Request ga...	90		27958	121	57	3
11	14:51:00.540	Thread Group 1-2	HTTP Request ga...	75		12345	121	29	1
2	14:51:00.134	Thread Group 1-1	HTTP Request ga...	84		12345	121	27	0
6	14:51:00.362	Thread Group 1-1	HTTP Request ga...	63		12345	121	21	0
3	14:51:00.218	Thread Group 1-1	HTTP Request ga...	44		27958	121	17	0
14	14:51:00.708	Thread Group 1-2	HTTP Request ga...	65		12345	121	17	0
12	14:51:00.615	Thread Group 1-2	HTTP Request ga...	58		12345	121	16	0
4	14:51:00.262	Thread Group 1-1	HTTP Request ga...	61		12345	121	15	0
10	14:51:00.554	Thread Group 1-1	HTTP Request ga...	40		27951	121	14	0
18	14:51:00.899	Thread Group 1-2	HTTP Request ga...	56		12345	121	14	0
9	14:51:00.514	Thread Group 1-1	HTTP Request ga...	40		27951	121	13	0
5	14:51:00.323	Thread Group 1-1	HTTP Request ga...	39		27951	121	12	0
8	14:51:00.462	Thread Group 1-1	HTTP Request ga...	52		12345	121	12	0
13	14:51:00.673	Thread Group 1-2	HTTP Request ga...	35		27958	121	12	0
15	14:51:00.773	Thread Group 1-2	HTTP Request ga...	37		27951	121	12	0
16	14:51:00.810	Thread Group 1-2	HTTP Request ga...	53		12345	121	12	0
17	14:51:00.863	Thread Group 1-2	HTTP Request ga...	36		27958	121	12	0
19	14:51:00.955	Thread Group 1-2	HTTP Request ga...	39		27951	121	12	0
20	14:51:00.994	Thread Group 1-2	HTTP Request ga...	55		12353	121	12	0
7	14:51:00.426	Thread Group 1-1	HTTP Request ga...	35		27951	121	11	0



























Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency ↓	Connect Time(ms)
10	14:16:36.294	Thread Group 1-2	HTTP Request ga...	155		12345	121	36	1
1	14:16:35.792	Thread Group 1-1	HTTP Request ga...	79		12345	121	33	0
11	14:16:36.319	Thread Group 1-1	HTTP Request ga...	133		12345	121	24	0
12	14:16:36.449	Thread Group 1-2	HTTP Request ga...	68		12345	121	20	0
15	14:16:36.666	Thread Group 1-2	HTTP Request ga...	73		12345	121	17	0
14	14:16:36.586	Thread Group 1-2	HTTP Request ga...	80		12345	121	16	0
3	14:16:35.927	Thread Group 1-1	HTTP Request ga...	58		12345	121	15	0
2	14:16:35.871	Thread Group 1-1	HTTP Request ga...	56		12345	121	14	0
6	14:16:36.089	Thread Group 1-1	HTTP Request ga...	53		12345	121	14	0
16	14:16:36.739	Thread Group 1-2	HTTP Request ga...	57		12345	121	14	0
7	14:16:36.142	Thread Group 1-1	HTTP Request ga...	60		12345	121	13	0
8	14:16:36.202	Thread Group 1-1	HTTP Request ga...	53		12345	121	13	0
13	14:16:36.517	Thread Group 1-2	HTTP Request ga...	69		12345	121	13	0
19	14:16:36.902	Thread Group 1-2	HTTP Request ga...	51		12345	121	13	0
4	14:16:35.985	Thread Group 1-1	HTTP Request ga...	52		12345	121	12	0
5	14:16:36.037	Thread Group 1-1	HTTP Request ga...	52		12345	121	12	0
9	14:16:36.255	Thread Group 1-1	HTTP Request ga...	64		12345	121	12	0
17	14:16:36.796	Thread Group 1-2	HTTP Request ga...	52		12345	121	12	0
18	14:16:36.848	Thread Group 1-2	HTTP Request ga...	54		12345	121	12	0
20	14:16:36.954	Thread Group 1-2	HTTP Request ga...	51		12345	121	12	0



























Imagens das métricas capturadas pelo Apache JMeter (2 servidores / 1 servidor – 40 pedidos por segundo)

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency ↓	Connect Time(ms)
16	14:17:33.046	Thread Group 1-4	HTTP Request ga...	240		12345	121	49	2
11	14:17:32.796	Thread Group 1-3	HTTP Request ga...	176		12345	121	46	1
18	14:17:33.161	Thread Group 1-3	HTTP Request ga...	227		12345	121	41	0
20	14:17:33.283	Thread Group 1-1	HTTP Request ga...	242		12345	121	39	0
6	14:17:32.545	Thread Group 1-2	HTTP Request ga...	128		12345	121	36	1
19	14:17:33.286	Thread Group 1-4	HTTP Request ga...	206		12345	121	36	0
17	14:17:33.137	Thread Group 1-2	HTTP Request ga...	241		12345	121	34	0
22	14:17:33.388	Thread Group 1-3	HTTP Request ga...	184		12345	121	33	0
1	14:17:32.295	Thread Group 1-1	HTTP Request ga...	77		12345	121	30	1
15	14:17:33.056	Thread Group 1-1	HTTP Request ga...	227		12345	121	30	0
21	14:17:33.379	Thread Group 1-2	HTTP Request ga...	190		12345	121	30	0
31	14:17:33.884	Thread Group 1-2	HTTP Request ga...	149		12345	121	29	0
34	14:17:34.033	Thread Group 1-2	HTTP Request ga...	165		12345	121	29	0
25	14:17:33.572	Thread Group 1-3	HTTP Request ga...	158		12345	121	28	0
30	14:17:33.879	Thread Group 1-3	HTTP Request ga...	153		12345	121	28	0
27	14:17:33.730	Thread Group 1-3	HTTP Request ga...	149		12345	121	27	0
14	14:17:32.972	Thread Group 1-3	HTTP Request ga...	189		12345	121	26	0
23	14:17:33.492	Thread Group 1-4	HTTP Request ga...	158		12345	121	26	0
26	14:17:33.650	Thread Group 1-4	HTTP Request ga...	165		12345	121	26	0
29	14:17:33.816	Thread Group 1-4	HTTP Request ga...	159		12345	121	25	0
13	14:17:32.957	Thread Group 1-2	HTTP Request ga...	179		12345	121	24	0
28	14:17:33.726	Thread Group 1-2	HTTP Request ga...	158		12345	121	24	0
33	14:17:34.032	Thread Group 1-3	HTTP Request ga...	164		12345	121	24	0
12	14:17:32.896	Thread Group 1-1	HTTP Request ga...	160		12345	121	23	0
24	14:17:33.570	Thread Group 1-2	HTTP Request ga...	155		12345	121	23	0
32	14:17:33.975	Thread Group 1-4	HTTP Request ga...	161		12345	121	23	0

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency ↓	Connect Time(ms)
24	14:51:36.295	Thread Group 1-4	HTTP Request ga...	187		12345	121	46	1
1	14:51:35.547	Thread Group 1-1	HTTP Request ga...	69		27958	121	45	2
15	14:51:36.047	Thread Group 1-3	HTTP Request ga...	138		12345	121	39	2
6	14:51:35.797	Thread Group 1-2	HTTP Request ga...	78		12345	121	30	1
33	14:51:36.611	Thread Group 1-3	HTTP Request ga...	141		12345	121	30	0
2	14:51:35.616	Thread Group 1-1	HTTP Request ga...	77		12345	121	29	0
22	14:51:36.324	Thread Group 1-3	HTTP Request ga...	64		27958	121	26	0
25	14:51:36.380	Thread Group 1-2	HTTP Request ga...	140		12345	121	26	0
28	14:51:36.439	Thread Group 1-3	HTTP Request ga...	127		12345	121	25	0
32	14:51:36.563	Thread Group 1-4	HTTP Request ga...	145		12345	121	24	0
13	14:51:35.991	Thread Group 1-2	HTTP Request ga...	98		12345	121	22	0
11	14:51:35.940	Thread Group 1-1	HTTP Request ga...	88		12345	121	21	0
8	14:51:35.887	Thread Group 1-1	HTTP Request ga...	52		27951	121	20	0
12	14:51:36.028	Thread Group 1-1	HTTP Request ga...	52		27958	121	20	0
23	14:51:36.388	Thread Group 1-3	HTTP Request ga...	50		27958	121	20	0
14	14:51:36.080	Thread Group 1-1	HTTP Request ga...	43		27951	121	19	0
16	14:51:36.090	Thread Group 1-2	HTTP Request ga...	103		12345	121	19	0
30	14:51:36.520	Thread Group 1-2	HTTP Request ga...	145		12345	121	19	0
34	14:51:36.709	Thread Group 1-4	HTTP Request ga...	80		12345	121	19	0
7	14:51:35.843	Thread Group 1-1	HTTP Request ga...	44		27951	121	18	0
19	14:51:36.226	Thread Group 1-3	HTTP Request ga...	49		27951	121	18	0
26	14:51:36.482	Thread Group 1-4	HTTP Request ga...	42		27958	121	18	0
29	14:51:36.566	Thread Group 1-3	HTTP Request ga...	45		27951	121	18	0
5	14:51:35.797	Thread Group 1-1	HTTP Request ga...	46		27958	121	17	0
10	14:51:35.943	Thread Group 1-2	HTTP Request ga...	48		27958	121	17	0
4	14:51:35.732	Thread Group 1-1	HTTP Request ga...	65		12345	121	16	0

Imagens das métricas capturadas pelo Apache JMeter (2 servidores / 1 servidor – 80 pedidos por segundo)

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency ↓	Connect Time(ms)
41	14:20:38.448	Thread Group 1-1	HTTP Request ga...	499		12345	121	124	0
62	14:20:39.709	Thread Group 1-7	HTTP Request ga...	434		12345	121	117	0
61	14:20:39.659	Thread Group 1-2	HTTP Request ga...	444		12345	121	110	0
32	14:20:38.011	Thread Group 1-1	HTTP Request ga...	437		12345	121	91	0
15	14:20:37.137	Thread Group 1-7	HTTP Request ga...	432		12345	121	87	0
18	14:20:37.245	Thread Group 1-5	HTTP Request ga...	506		12345	121	80	0
19	14:20:37.264	Thread Group 1-8	HTTP Request ga...	501		12345	121	75	1
25	14:20:37.598	Thread Group 1-6	HTTP Request ga...	477		12345	121	73	0
16	14:20:37.146	Thread Group 1-1	HTTP Request ga...	429		12345	121	72	0
27	14:20:37.765	Thread Group 1-8	HTTP Request ga...	433		12345	121	72	0
20	14:20:37.136	Thread Group 1-3	HTTP Request ga...	712		12345	121	71	0
24	14:20:37.575	Thread Group 1-1	HTTP Request ga...	436		12345	121	71	0
54	14:20:39.262	Thread Group 1-2	HTTP Request ga...	397		12345	121	70	0
23	14:20:37.569	Thread Group 1-7	HTTP Request ga...	434		12345	121	68	0
31	14:20:38.004	Thread Group 1-7	HTTP Request ga...	433		12345	121	68	0
47	14:20:38.846	Thread Group 1-7	HTTP Request ga...	418		12345	121	67	0
39	14:20:38.438	Thread Group 1-7	HTTP Request ga...	408		12345	121	66	0
58	14:20:39.454	Thread Group 1-3	HTTP Request ga...	476		12345	121	66	0
22	14:20:37.528	Thread Group 1-2	HTTP Request ga...	449		12345	121	65	0
55	14:20:39.264	Thread Group 1-7	HTTP Request ga...	445		12345	121	65	0
7	14:20:36.639	Thread Group 1-3	HTTP Request ga...	217		12345	121	64	1
17	14:20:37.013	Thread Group 1-6	HTTP Request ga...	585		12345	121	62	1
29	14:20:37.977	Thread Group 1-2	HTTP Request ga...	416		12345	121	62	0
26	14:20:37.751	Thread Group 1-5	HTTP Request ga...	437		12345	121	60	0
30	14:20:37.966	Thread Group 1-4	HTTP Request ga...	431		12345	121	60	0
69	14:20:40.209	Thread Group 1-6	HTTP Request ga...	328		12345	121	60	0

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency ↓	Connect Time(ms)
19	14:52:33.921	Thread Group 1-5	HTTP Request ga...	402		12345	121	76	1
28	14:52:34.167	Thread Group 1-7	HTTP Request ga...	432		12345	121	73	1
58	14:52:35.121	Thread Group 1-8	HTTP Request ga...	410		12345	121	73	0
27	14:52:34.122	Thread Group 1-2	HTTP Request ga...	440		12345	121	71	0
44	14:52:34.689	Thread Group 1-7	HTTP Request ga...	402		12345	121	66	0
15	14:52:33.794	Thread Group 1-4	HTTP Request ga...	332		12345	121	65	0
40	14:52:34.541	Thread Group 1-6	HTTP Request ga...	454		12345	121	65	0
38	14:52:34.479	Thread Group 1-5	HTTP Request ga...	411		12352	121	64	0
64	14:52:35.361	Thread Group 1-5	HTTP Request ga...	374		12345	121	64	0
29	14:52:34.229	Thread Group 1-4	HTTP Request ga...	408		12345	121	63	0
32	14:52:34.296	Thread Group 1-8	HTTP Request ga...	419		12345	121	61	1
25	14:52:34.046	Thread Group 1-6	HTTP Request ga...	445		12345	121	59	0
41	14:52:34.637	Thread Group 1-4	HTTP Request ga...	403		12345	121	58	0
71	14:52:35.617	Thread Group 1-7	HTTP Request ga...	301		12345	121	56	0
34	14:52:34.341	Thread Group 1-3	HTTP Request ga...	430		12345	121	54	0
1	14:52:33.429	Thread Group 1-1	HTTP Request ga...	84		27958	121	52	11
22	14:52:34.005	Thread Group 1-1	HTTP Request ga...	378		12345	121	52	0
35	14:52:34.383	Thread Group 1-1	HTTP Request ga...	422		12345	121	51	0
52	14:52:34.941	Thread Group 1-5	HTTP Request ga...	380		12352	121	51	0
65	14:52:35.308	Thread Group 1-3	HTTP Request ga...	432		12345	121	51	0
67	14:52:35.456	Thread Group 1-6	HTTP Request ga...	356		12345	121	51	0
18	14:52:33.903	Thread Group 1-3	HTTP Request ga...	364		12345	121	49	0
50	14:52:34.851	Thread Group 1-3	HTTP Request ga...	413		12345	121	49	0
55	14:52:35.085	Thread Group 1-6	HTTP Request ga...	371		12345	121	48	0
62	14:52:35.229	Thread Group 1-1	HTTP Request ga...	393		12352	121	48	0
45	14:52:34.757	Thread Group 1-8	HTTP Request ga...	363		12345	121	47	0