

DOI: 10.5748/19CONTECSI/PSE/ISM/7000

**ESCALABILIDADE HORIZONTAL COM CLUSTER DE ALTA
DISPONIBILIDADE COM BALANCEAMENTO DE CARGA EM MÁQUINAS
VIRTUAIS**

Samuel Antonio Vieira ; <https://orcid.org/0000-0002-1408-4183>
Faculdade de Tecnologia de Tatuí



HORIZONTAL SCALABILITY WITH HIGH AVAILABILITY CLUSTER WITH LOAD BALANCE IN VIRTUAL MACHINES

ABSTRACT

Clustering is a common issue within data centers. Information is one of the most valuable assets nowadays and data is the raw material for this. In this way, access to information needs to be always available and secure. Security is paramount, from its level of access, to its availability, and availability is what this article is about. Data centers work with redundancy and load balancing systems and employ this to offer a better service in the cloud, On-Premises systems can also offer specific services in the same way. When processing reaches the hardware limit, new resources are needed to keep the system online. Scheduling is the resource used by Information Technology professionals to increase processing power and keep resources available. MariaDB is an open source database that offers clustering through Galera Cluster, with a very complete documentation on the developer's website and it will be shown in a practical way how to install and configure a high availability cluster with load balancing in virtual machines on Desktop hardware. This article was born as an arm of the research that the author develops in the institution of Higher Education, Faculdade de Tecnologia de Tatuí-SP - Prof. Wilson R. R. de Camargo whose title is "Using free software for server clustering and virtualization".

Keywords: Linux. Clustering. High Availability. Load Balance. Horizontal Scalability.

ESCALABILIDADE HORIZONTAL COM CLUSTER DE ALTA DISPONIBILIDADE COM BALANCEAMENTO DE CARGA EM MÁQUINAS VIRTUAIS

RESUMO

A clusterização é um assunto comum dentro dos data centers. A informação é um dos bens mais valiosos nos dias atuais e os dados são a matéria prima para tal. Desta forma o acesso a informação precisa estar sempre disponível e em segurança. A segurança é primordial, desde o seu nível de acesso, até a sua disponibilidade e é na disponibilidade que este artigo fala. Os data centers trabalham com sistemas de redundância e balanceamento de carga e empregam isso para oferecer um serviço melhor na nuvem, sistemas On-Premises também podem oferecer serviços específicos da mesma forma. Quando o processamento atinge o limite do hardware, novos recursos são necessários para manter o sistema on-line. A escalabilidade é o recurso utilizado por profissionais da Tecnologia da Informação para aumentar o poder de processamento e manter os recursos disponíveis. O MariaDB é um banco de dados de código aberto que oferece uma clusterização através do Galera Cluster, com uma documentação bastante completa no site do desenvolvedor e será mostrado de forma prática a sua instalação e configuração de um cluster de alta disponibilidade com balanceamento de carga em máquinas virtuais em um hardware Desktop. Este artigo nasceu como um braço da pesquisa que o autor desenvolve na instituição de Ensino Superior, a Faculdade de Tecnologia de Tatuí-SP - Prof. Wilson R. R. de Camargo cujo título é "Utilização de software livre para clusterização e virtualização de servidores".

Palavras-Chave: Linux. Cluster. Alta Disponibilidade. Balanceamento de Carga. Escalabilidade.

INTRODUÇÃO

Cluster é um assunto bastante discutido dentro do mundo da Tecnologia da Informação. É comum sistemas em nuvem utilizam esta tecnologia para segurança, disponibilidade, poder de processamento e armazenamento, tudo empregado em seus serviços: Software como Serviço, Plataforma como Serviço e Infraestrutura como Serviço.

Pitanga (2003) define este aglomerado de computadores como uma forma de fornecer acesso aos usuários sem a percepção do uso de múltiplas máquinas. Cada computador (ou nó) trabalha de forma síncrona (ou assíncrona) para executar as tarefas em conjunto.

O assunto pode parecer complexo e longe da realidade de empresas que optam por sistemas On-Premises. Isto pode dar a impressão de que a clusterização não se aplica a esta tecnologia, ou que ela está atrelada a sistemas de alto-desempenho dentro de um *Cluster Bewolf*, ou que uma clusterização tende a fazer mais sentido utilizando máquinas reais.

Uma das barreiras vencidas em servidores locais, para pequenas e médias empresas foi o uso da virtualização, de forma a quebrar o paradigma de todos os serviços serem disponibilizados em conjunto. Desta forma, o uso da virtualização se tornou muito popular por trazer o isolamento dos serviços nos servidores.

Com base nisso, o objetivo deste trabalho é demonstrar um cluster de alta-disponibilidade com o sistema gerenciador de banco de dados MariaDB, dentro de máquinas virtuais com sistema Oracle VirtualBox e sua escalabilidade. Ele é um complemento do trabalho de Vieira (2022), onde demonstrou toda a instalação de um cluster de Alta-Disponibilidade com Balanceamento de Carga em máquinas virtuais, utilizando o Oracle VirtualBox. Como objetivo específico será feito o processo de escalabilidade horizontal deste cluster.

O trabalho visa demonstrar que a clusterização em sistemas virtuais são fáceis de escalar, de forma a desmistificar seu uso. A clusterização será feita diretamente nos hosts convidados.

CLUSTER, VIRTUALIZAÇÃO E ESCALABILIDADE

Para Perez (2021) os clusters ganharam força na década de 90, com a depreciação da escala vertical de recursos. Este tipo de escalabilidade é característico pelo upgrade de hardware dos servidores. Desta forma pode-se aumentar o poder de processamento físico, aumento de disponibilidade de memória RAM e/ou aumento de armazenamento em disco (Escola Superior de Redes, 2021). A clusterização não é um assunto novo, seu surgimento se deve à década de 60, quando a Industry Business Machine (IBM), interligou seus Mainframes para obter processamento paralelo. Sua função era aumentar a eficiência e oferecer viabilidade neste tipo de processamento.

Com a virada do milênio e a massificação da Internet, houve uma necessidade de aumento de desempenho maior que a capacidade de escalabilidade do hardware. Nessa época surgiu a escala horizontal de recursos. A cada nova necessidade de aumento de desempenho, um novo host era adicionado ao sistema. O software open source teve um papel importante para este desenvolvimento. (PEREZ, 2021)

Define-se escalabilidade vertical como o aumento de recursos de forma a balancear os serviços disponíveis, para não sobrecarregar o processamento do hardware. Um movimento muito comum na Cloud Computing.

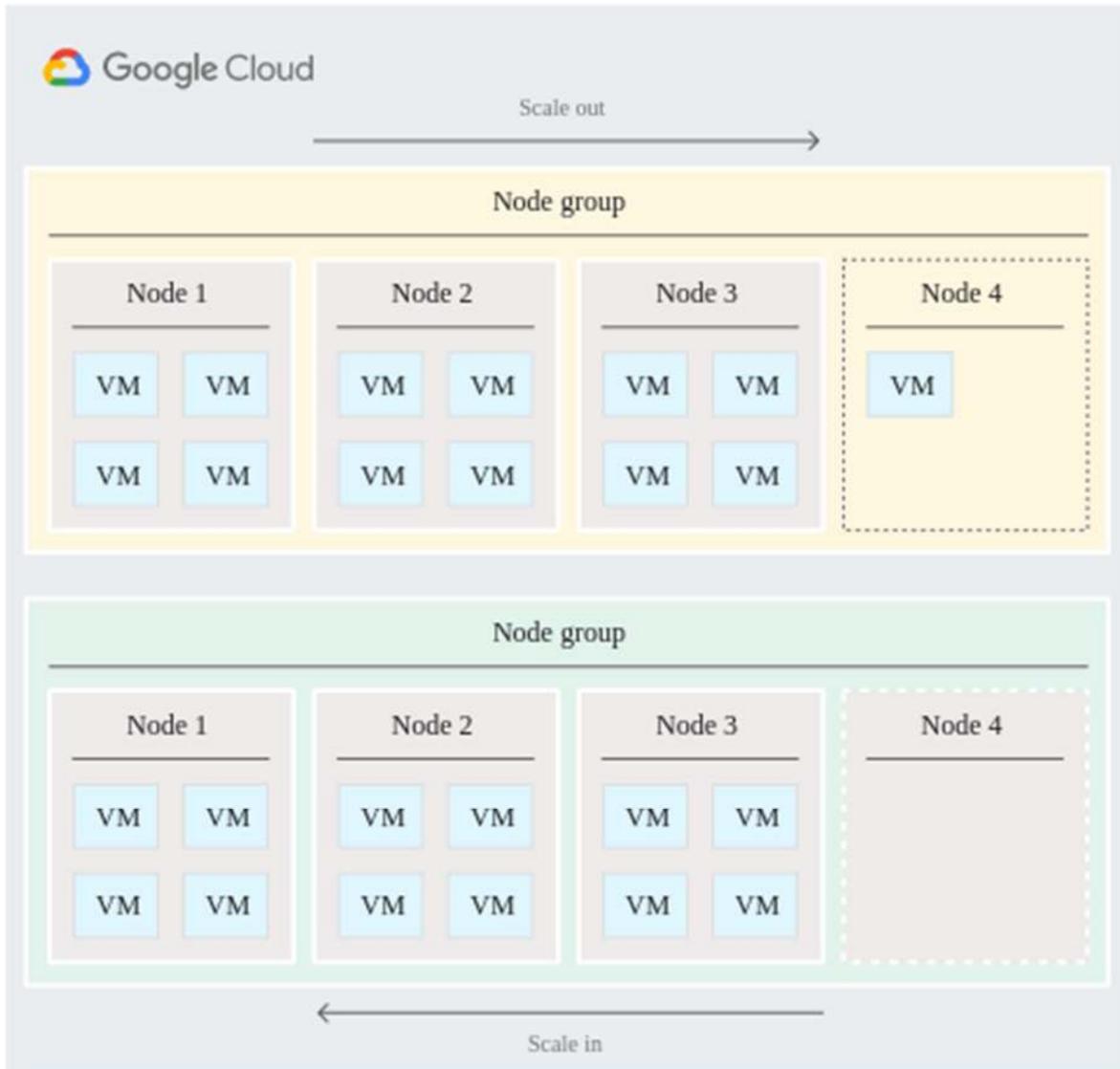


Figura 1: Escalabilidade. Fonte: <https://cloud.google.com/compute/docs/nodes/autoscaling-node-groups?hl=pt-br#scale_out_and_scale_in> Acesso em: 15 out. 2022

A figura demonstra o escalabilidade de máquinas virtuais e seus nós, dentro uma estrutura do Google Cloud. Nela pode-se notar o escalabilidade vertical, com o aumento ou diminuição de nós e o escalabilidade horizontal com o aumento ou diminuição de máquinas virtuais (VM).

Nós (nodos) é como são chamados os hosts de um cluster. A requisição é enviada e processada, de acordo com o tipo de cluster empregado, no lado cliente, o usuário não se dá conta de qual tipo de cluster está inserido ou se está inserido. Por exemplo, quando se faz uma consulta do Google, a requisição vai para o seu sistema e é processado pelo seu data center, o usuário não tem controle para onde foi a requisição. (PITANGA, 2003)

O autor afirma que os clusters estão divididos em:

- Alta disponibilidade (High Availability (HA) and Failover): são aqueles que visam oferecer um serviço ininterrupto. Caso haja falha em algum nó outro assumirá a tarefa

de disponibilizar o recurso e/ou serviço, é muito comum encontrá-los no modelo nó-mestre e nó(s)-escravo(s);

- Balanceamento de carga (Load Balancing): neste tipo de cluster, trabalha-se com o paradigma nó-mestre-nó-mestre, a cada requisição recebida, o cluster envia ao nó disponível. Outra familiaridade é que este tipo de cluster trabalha com a combinação HA e Load Balancing.
- Processamento Distribuído ou Processamento Paralelo: este visa melhorar o desempenho geral de processamento do sistema. As aplicações rodam de forma distribuída nos nodos, formando um supercomputador.
- Cluster Beowulf: Este cluster está empregado dentro dos sistemas de Processamento Distribuído, ele foi implementado por Thomas Sterling e Donald Becker em 1994. Sua peculiaridade é usar máquinas de baixo desempenho para criar um supercomputador para uso científico.

Pitanga (2003) afirma que a potencialização desse tipo sistema computacional pode aumentar a segurança, o desempenho e reduzir os custos, junto com Perez (2021) corrobora com a ideia onde a inclusão de máquinas de uso cotidiano pode-se criar máquinas com grande capacidade de processamento, sem a necessidade de hardwares padronizados ou especializados. Por fim, devido a essa heterogeneidade sistêmica, é possível uma escalabilidade, aumentando a quantidade de nós devido às demandas vindouras.

O VirtualBox é uma ferramenta de virtualização que pertence à Oracle. Trabalha sob a licença GNU General Public License (GPL) versão 2. Este sistema suporta os principais sistemas operacionais de mercado como hospedeiros e convidados. (ORACLE, 2022) “Virtualização é uma tecnologia que permite criar serviços de TI valiosos usando recursos que estão tradicionalmente vinculados a um determinado hardware.” (REDHAT, 2018). Desta forma, pode-se dimensionar os serviços em VMs isolados. Graças aos hypervisors, o hardware é particionado virtualmente, disponibilizando-os aos convidados de acordo com a necessidade de cada.

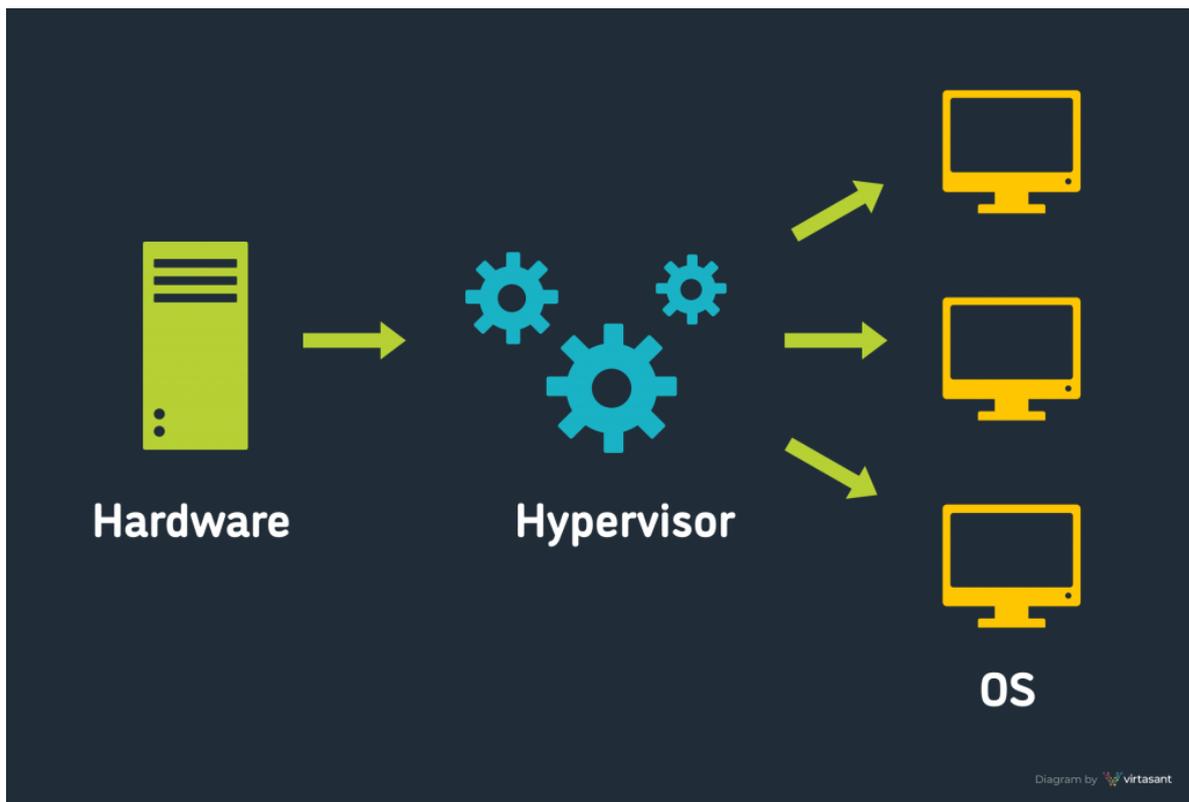


Figura 2: Hypervisors. Disponível em: <<https://www.virtasant.com/blog/hypervisors-a-comprehensive-guide>>. Acesso em: 19 set. 2022.

O hypervisor é responsável pelo controle e monitoramento das VMs, é responsável por gerenciar os recursos do hardware, como processamento e memória. (HAJDARBEGOVIC, 2020)

AMBIENTE INICIAL DO CLUSTER

Vieira (2022), utilizou os seguintes recursos para a montagem e configuração do cluster. Para a máquina hospedeira, foi utilizado o computador:

- Intel(R) Core (TM) i3 CPU 550 @ 3.20GHz,
- 4GB Ram,
- 500 GB de armazenamento,
- Sistema Debian 5.10.136-1 (2022-08-13) x86_64 GNU/Linux,
- Com virtualizador:
 - Oracle Virtual Box Versão 6.1.36 r152435 (Qt5.15.2),
- Para cada máquinas virtuais (VM):
 - Um núcleo,
 - 1GB de Ram e 20GB de armazenamento,
 - Sistema Operacional: Operating System: Debian GNU/Linux 11 (bullseye)
 - Kernel: Linux 5.10.0-17-amd64
 - Architecture: x86-64

Para a versão do MariaDB instalado: 10.5.15-MariaDB-0+deb11u1 for debian-linux-gnu on x86_64 (Debian 11), a replicação ficou por conta do galera-4, 26.4.11-0+deb11u1 em ambos os nodos denominados: db-node01 e db-node02.

O balanceamento de carga foi utilizado o HAProxy, versão: 2.2.9-2+deb11u3

INSTALAÇÃO

Vieira (2022) fez a instalação padrão do sistema, sem interface gráfica, utilizando todo o espaço disponível¹. As partições de disco no hospedeiro ficaram:

```
Senha:
root@host1:~# fdisk -l
Disco /dev/sda: 465,76 GiB, 500107862016 bytes, 976773168 setores
Modelo de disco: SAMSUNG HD502HJ
Unidades: setor de 1 * 512 = 512 bytes
Tamanho de setor (lógico/físico): 512 bytes / 512 bytes
Tamanho E/S (mínimo/ótimo): 512 bytes / 512 bytes
Tipo de rótulo do disco: dos
Identificador do disco: 0xcbad9b45

Dispositivo Inicializar      Início      Fim      Setores Tamanho Id Tipo
/dev/sda1      *              2048 974772223 974770176 464,8G 83 Linux
/dev/sda2              974774270 976771071 1996802    975M  5 Estendida
/dev/sda5              974774272 976771071 1996800    975M  82 Linux swap / So
```

Figura 3: Partições do disco: Fonte: Vieira, 2022

Nos convidados ficaram:

```
root@db-node01:~# fdisk -l
Disco /dev/sda: 19,53 GiB, 20971520000 bytes, 40960000 setores
Modelo de disco: VBOX HARDDISK
Unidades: setor de 1 * 512 = 512 bytes
Tamanho de setor (lógico/físico): 512 bytes / 512 bytes
Tamanho E/S (mínimo/ótimo): 512 bytes / 512 bytes
Tipo de rótulo do disco: dos
Identificador do disco: 0x75907776

Dispositivo Inicializar      Início      Fim      Setores Tamanho Id Tipo
/dev/sda1      *              2048 37750783 37748736    18G  83 Linux
/dev/sda2              37750784 40959999 3209216    1,5G  5 Estendida
/dev/sda5              37752832 40959999 3207168    1,5G  83 Linux
root@db-node01:~# █
```

Figura 4: Partições do disco VMs. Fonte: Vieira, 2022

Em ambos, não foi criada nenhuma outra partição.

Para as máquinas virtuais, foi criada um padrão, que foi clonada da seguinte forma:

- Uma máquina com sistema Linux;
- No clone desta primeira máquina, foram instalados o MariaDB e o Galera Cluster;
- A segunda máquina foi clonada integralmente;

¹ Artigo que será publicado na LatinScience em novembro de 2022.

- A primeira máquina foi instalada o HA Proxy após ela ser clonada (sem o MariaDB instalado)

Nas máquinas virtuais com banco de dados, foram instalados o mariadb-server-10.5 e o galera-4, após a instalação foi feita a configuração básica no nó.

Nó 1 (db-node01):

```
[galera]
wsrep_on=ON
wsrep_provider=/usr/lib/galera/libgalera_smm.so
wsrep_cluster_address="gcomm://db-node01,db-node02"
binlog_format=row
default_storage_engine=InnoDB
innodb_autoinc_lock_mode=2
bind-address=0.0.0.0
wsrep_cluster_name="galera_cluster"
wsrep_node_address="db-node01"
```

Arquivo: /etc/mysql/mariadb.conf.d/60-galera.cnf. Fonte: Vieira, 2022

Nó 2 (db-node02):

```
[galera]
wsrep_on=ON
wsrep_provider=/usr/lib/galera/libgalera_smm.so
wsrep_cluster_address="gcomm://db-node01,db-node02"
binlog_format=row
default_storage_engine=InnoDB
innodb_autoinc_lock_mode=2
bind-address=0.0.0.0
wsrep_cluster_name="galera_cluster"
wsrep_node_address="db-node02"
```

Arquivo: /etc/mysql/mariadb.conf.d/60-galera.cnf. Fonte: Vieira, 2022

Com a configuração habilitada, o cluster, configurou-se a biblioteca a ser utilizada, quais os hosts que iriam parte do cluster, o nível de replicação, o tipo de banco de dados, o método de trava de arquivos, quais endereços ips o banco vai responder, o nome do cluster e o nome do nó no cluster. Após a configuração, o sistema de banco de dados foi iniciado.

No nó balanceador, foi instalado o HA-Proxy e feito a configuração (db-servidor):

```
#-----
# Global settings
#-----
global
log      127.0.0.1 local2
chroot  /var/lib/haproxy
pidfile  /var/run/haproxy.pid
maxconn  4000
user     haproxy
```

```

group    haproxy
daemon
# turn on stats unix socket
stats socket /var/lib/haproxy/stats

defaults
mode     tcp
log      global
option   dontlognull
option   redispatch
retries  3
timeout queue 45s
timeout connect 5s
timeout client 1m
timeout server 1m
timeout check 10s
maxconn  3000

#-----
# main frontend which proxys to the backends
#-----
frontend main
bind 10.132.229.1:3306
default_backend app
#-----
# round robin balancing between the various backends
#-----
backend app
balance roundrobin
server app1 10.132.231.1:3306 maxconn 151 check
server app2 10.132.231.2:3306 maxconn 151 check

```

Arquivo: /etc/haproxy/haproxy.cf. Fonte: Vieira, 2022.

Com todos os serviços funcionando a estrutura ficou conforme a figura 5:

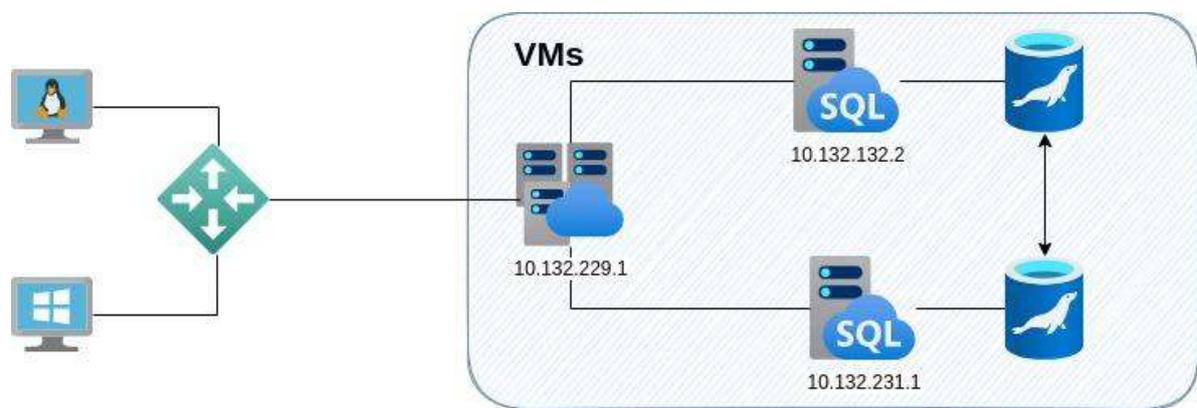


Figura 5: Estrutura Inicial do Cluster. Fonte: Vieira, 2022.

A figura 5 demonstra como ficaram organizadas as máquinas virtuais (VMs). Duas máquinas com MariaDB e o Galera Cluster trocam informações dentro de uma Alta

Disponibilidade Master-Master, respectivamente db-node01 (10.132.231.1) e db-node02 (10.132.231.2). O acesso será feito pela máquina chamada db-servidor (10.132.229.1) que fará o balanceamento das requisições.

Dentro do Oracle Virtual Box, as máquinas ficaram conforme a figura 6:

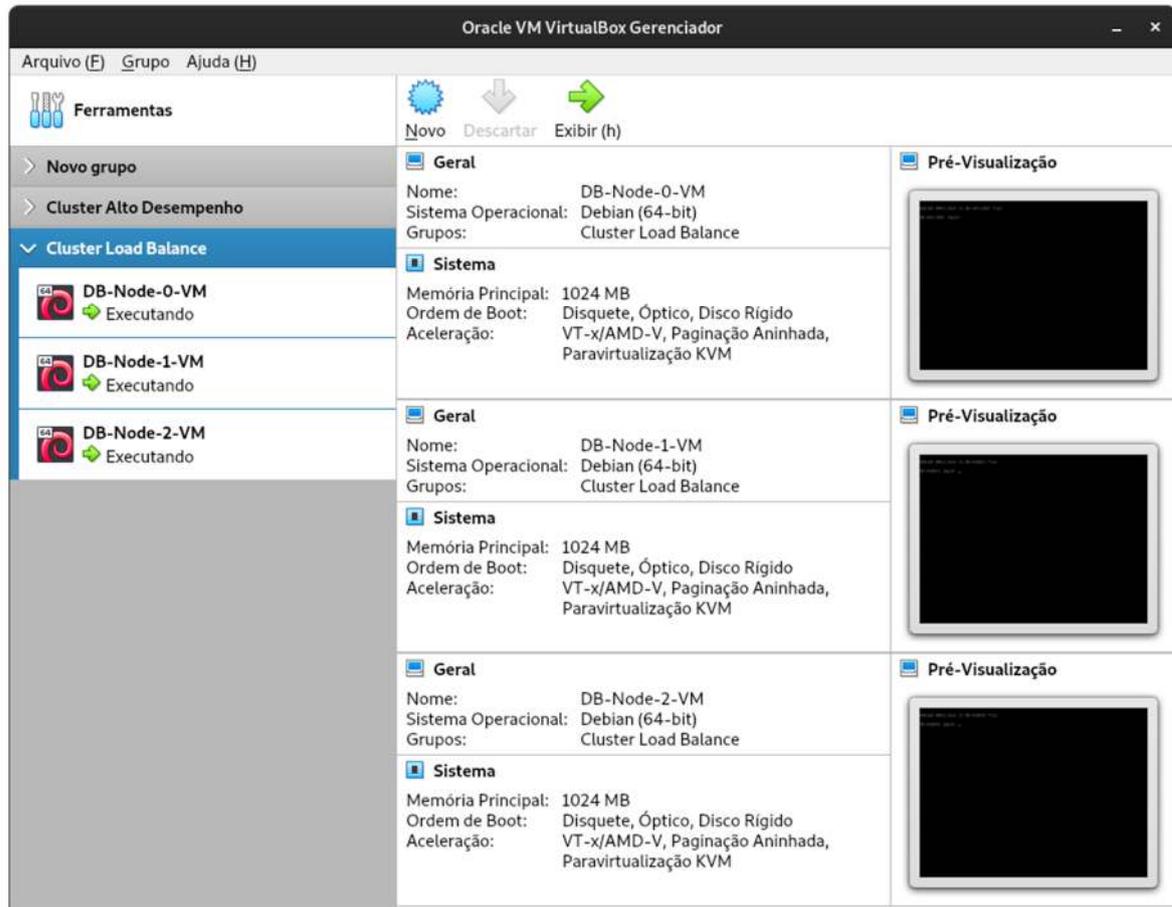


Figura 6: Organização Virtual Box. Fonte: Vieira, 2022.

A figura 6 ilustra a organização do cluster dentro do aplicativo, o grupo criado para aninhar as máquinas, chamado Cluster Load Balance, possui três VMs: DB-Node-0-VM (db-servidor) que receberá as requisições e fará a distribuição, DB-Node-1-VM (db-node01) e DB-Node-2-VM (db-node02) que são as máquinas virtuais com o MariaDB instalado. Esta organização também facilita a Escalabilidade Horizontal, neste caso, se necessário, clona-se a VM, configura e adiciona ao cluster.

Para verificar o número de nodos no cluster, pode se utilizar o comando abaixo:

```
samuel@samuel-pc: $ mysql -u $u -p$S -h 10.132.229.1 -D db_cluster -e "SHOW STATUS LIKE 'wsrep_cluster_size';"
mysql: [Warning] Using a password on the command line interface can be insecure.
+-----+-----+
| Variable_name | Value |
+-----+-----+
| wsrep_cluster_size | 2 |
+-----+-----+
```

Figura 7: Número de nodos. Fonte: Vieira, 2022.

A variável ambiente `wsrep_cluster_size` mostra a quantidade de nodos integrado ao cluster. Para a execução dos comandos, por questões de segurança, o usuário e senha do banco de dados foram exportados como variável ambiente dentro da sessão, ao fim dos testes, essas variáveis foram apagadas.

ACRESCENTANDO RECURSOS E ESCALONANDO O CLUSTER

Neste ponto a pesquisa avança aumentando os recursos empregados no cluster supracitado, foi instalado o mesmo sistema no host hospedeiro, de forma estruturar o cluster conforme a figura.8.

O `host0` é o hospedeiro do cluster que que fora iniciado, com duas máquinas virtuais com banco de dados e uma máquina virtual para fazer o balanceamento de carga. O `host1` será o hospedeiro acrescentado com três máquinas virtuais com banco de dados. A instalação do sistema operacional segue os mesmos parâmetros do `host0`:

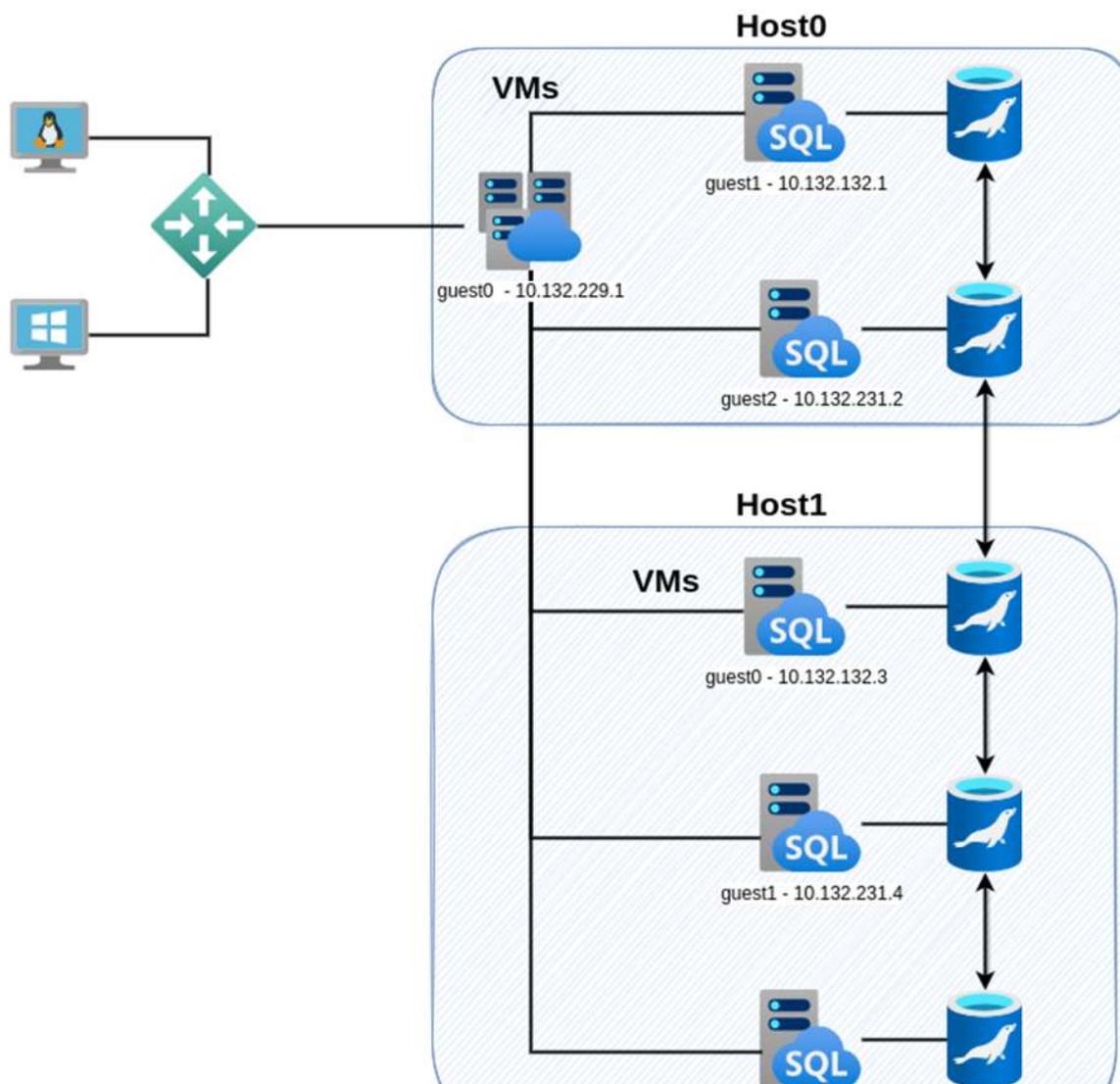


Figura 8: Cluster escalonado. Fonte: Elaboração Própria

- Intel(R) Core (TM) i3 CPU 550 @ 3.20GHz,
- 4GB Ram,
- 500 GB de armazenamento,
- Sistema Debian 5.10.136-1 (2022-08-13) x86_64 GNU/Linux,
- Com virtualizador:
 - Oracle Virtual Box Versão Versão 7.0.0 r153978 (Qt5.15.2),
- Para cada máquina virtual (VM):
 - Um núcleo,
 - 1GB de Ram e 20GB de armazenamento,
 - Sistema Operacional: Operating System: Debian GNU/Linux 11 (bullseye)
 - Kernel: Linux 5.10.0-17-amd64
 - Architecture: x86-64

Para a versão do MariaDB instalado: 10.5.15-MariaDB-0+deb11u1 for debian-linux-gnu on x86_64 (Debian 11), a replicação ficou por conta do galera-4, 26.4.11-0+deb11u1 em todos os nodos (novos e antigos), onde os novos foram denominados: db-node03, db-node04 e db-node05. A exceção das versões dos softwares instalados fica por conta do Oracle Virtual Box que lançou uma nova versão de seu sistema: Versão 7.0.0 r153978 (Qt5.15.2).

Os novos nodos foram clonados do primeiro host, ficando a disposição do host1 da seguinte forma:

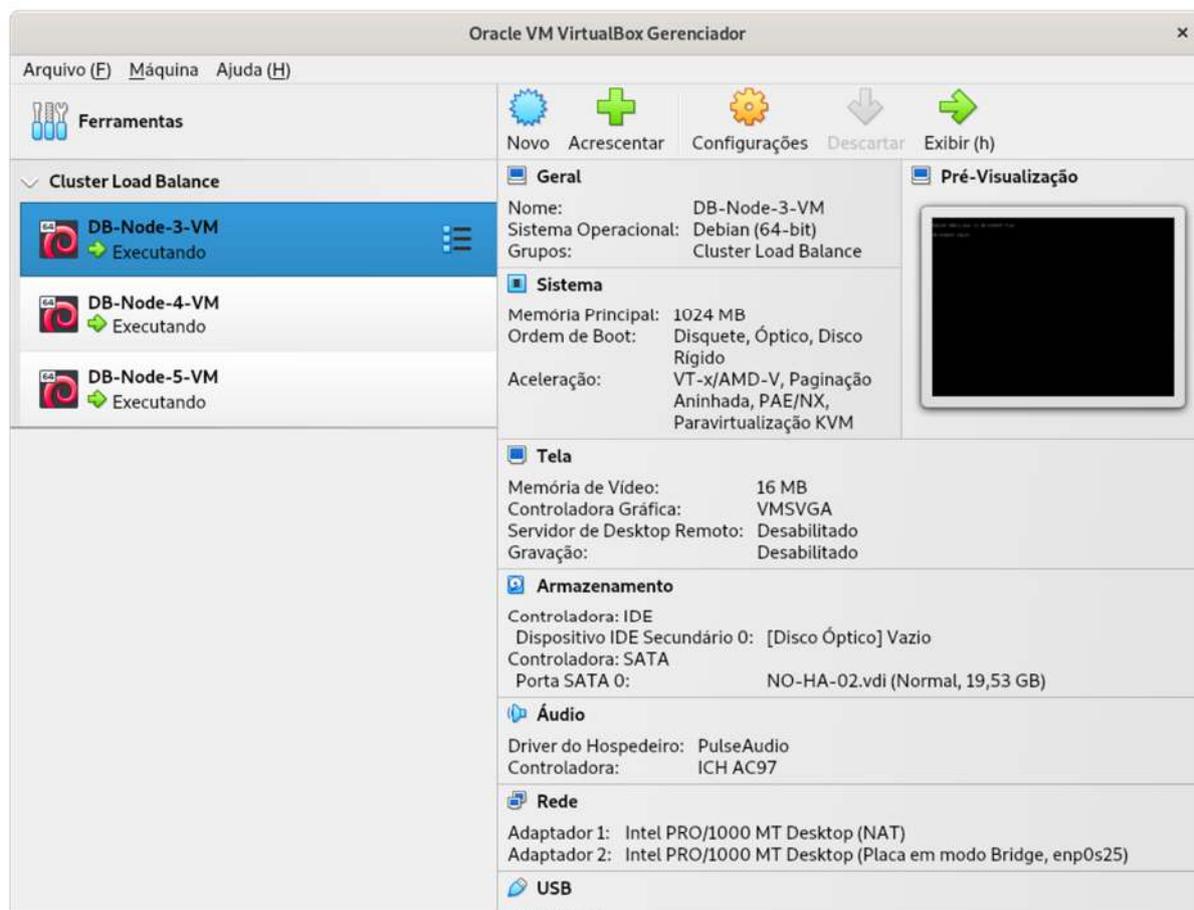


Figura 9: Organização Virtual Box host1. Fonte: Elaboração Própria.

As configurações de todos os nodos precisaram ser atualizadas.No arquivo /etc/hosts o grupo nós com seus nomes e endereços IPs ficaram com a seguinte distribuição em todas as máquinas virtuais:

```
10.132.229.1 db-servidor
10.132.231.1 db-node01
10.132.231.2 db-node02
10.132.231.3 db-node03
10.132.231.4 db-node04
10.132.231.5 db-node05
```

Para o escalabilidade horizontal, foi confirmado o arquivo /etc/mysql/mariadb.conf.d/60-galera.cnf nos novos nodos e atualizados os anteriores para que pudessem reconhecer os novos recursos adicionados ao cluster:

Nó 1 (db-node01) - alterado:

```
[galera]
wsrep_on=ON
wsrep_provider=/usr/lib/galera/libgalera_smm.so
wsrep_cluster_address="gcomm://db-node01,db-node02, db-node03,db-node04,db-node05"
binlog_format=row
default_storage_engine=InnoDB
innodb_autoinc_lock_mode=2
bind-address=0.0.0.0
wsrep_cluster_name="galera_cluster"
wsrep_node_address="db-node01"
```

Arquivo: /etc/mysql/mariadb.conf.d/60-galera.cnf. Fonte: Autoria Própria.

Nó 2 (db-node02) - alterado:

```
[galera]
wsrep_on=ON
wsrep_provider=/usr/lib/galera/libgalera_smm.so
wsrep_cluster_address="gcomm://db-node01,db-node02, db-node03,db-node04,db-node05"
binlog_format=row
default_storage_engine=InnoDB
innodb_autoinc_lock_mode=2
bind-address=0.0.0.0
wsrep_cluster_name="galera_cluster"
wsrep_node_address="db-node02"
```

Arquivo: /etc/mysql/mariadb.conf.d/60-galera.cnf. Fonte: Autoria Própria.

Nó 3 (db-node03) - adicionado:

```
[galera]
wsrep_on=ON
wsrep_provider=/usr/lib/galera/libgalera_smm.so
wsrep_cluster_address="gcomm://db-node01,db-node02, db-node03,db-node04,db-node05"
binlog_format=row
default_storage_engine=InnoDB
```

```
innodb_autoinc_lock_mode=2
bind-address=0.0.0.0
wsrep_cluster_name="galera_cluster"
wsrep_node_address="db-node03"
```

Arquivo: /etc/mysql/mariadb.conf.d/60-galera.cnf. Fonte: Aatoria Própria.

Nó 4 (db-node04) - adicionado:

```
[galera]
wsrep_on=ON
wsrep_provider=/usr/lib/galera/libgalera_smm.so
wsrep_cluster_address="gcomm://db-node01,db-node02, db-node03,db-node04,db-node05"
binlog_format=row
default_storage_engine=InnoDB
innodb_autoinc_lock_mode=2
bind-address=0.0.0.0
wsrep_cluster_name="galera_cluster"
wsrep_node_address="db-node04"
```

Arquivo: /etc/mysql/mariadb.conf.d/60-galera.cnf. Fonte: Aatoria Própria.

Nó 5 (db-node05) - adicionado:

```
[galera]
wsrep_on=ON
wsrep_provider=/usr/lib/galera/libgalera_smm.so
wsrep_cluster_address="gcomm://db-node01,db-node02, db-node03,db-node04,db-node05"
binlog_format=row
default_storage_engine=InnoDB
innodb_autoinc_lock_mode=2
bind-address=0.0.0.0
wsrep_cluster_name="galera_cluster"
wsrep_node_address="db-node05"
```

Arquivo: /etc/mysql/mariadb.conf.d/60-galera.cnf. Fonte: Aatoria Própria.

Com o cluster de alta disponibilidade escalonado, foi reiniciado/rearregado os serviços do MariaDB para sincronismo. Como os sistemas foram clonados das máquinas já existentes, não houve processo de sincronismo do seu início, o banco fez suas verificações e manteve os dados.

```
samuel@samuel-pc:~$ mysql -u $u -p$s -h 10.132.229.1 -D db_cluster -e "SHOW STATUS LIKE 'wsrep_cluster_size';"
mysql: [Warning] Using a password on the command line interface can be insecure.
+-----+-----+
| Variable_name | Value |
+-----+-----+
| wsrep_cluster_size | 5 |
+-----+-----+
```

Figura 10: Novos nodos. Fonte: Elaboração Própria.

A figura demonstra a quantidade de nodos ativos no cluster, outrora eram dois nodos (Figura X) agora são cinco nodos ativos.

O sistema balanceador de carga também sofreu alteração na sua configuração `/etc/haproxy/haproxy.cfg`, para que ele pudesse entender os novos nodos como disponíveis para receber requisições, conforme o trecho:

```
backend app
  balance roundrobin
  server app1 10.132.231.1:3306 maxconn 151 check
  server app2 10.132.231.2:3306 maxconn 151 check
  server app3 10.132.231.3:3306 maxconn 151 check
  server app4 10.132.231.4:3306 maxconn 151 check
  server app5 10.132.231.5:3306 maxconn 151 check
```

Foram adicionados novos nodos na porta 3306, padrão do MariaDB com a mesma quantidade de conexões. Reiniciado/recarregado o serviço de balanceamento de carga, os testes podem ser feitos.

METODOLOGIA

Esta pesquisa é um desenvolvimento experimental, com aplicação prática do escalonamento horizontal em e um cluster de Alta Disponibilidade com Balanceamento de Carga dentro do modelo Master-Master. As máquinas empregadas para isso são:

- Intel(R) Core (TM) i3 CPU 550 @ 3.20GHz,
- 4GB Ram, 500 GB de armazenamento,
- Sistema Debian 5.10.136-1 (2022-08-13) x86_64 GNU/Linux.
- Com virtualizador Oracle Virtual Box:
 - 6.1.36 r152435 (Qt5.15.2) e
 - 7.0.0 r153978 (Qt5.15.2).

Para cada máquina virtual, foi dimensionado da seguinte forma:

- Um núcleo,
- 1GB de Ram e
- 20GB de armazenamento,
- Sistema Operacional: Operating System: Debian GNU/Linux 11 (bullseye),
- Kernel: Linux 5.10.0-17-amd64,
- Architecture: x86-64.

Após a instalação básica do sistema, foram instalados os softwares de banco de dados MariaDB (10.5.15-MariaDB-0+deb11u1), GaleraCluster (galera-4, 26.4.11-0+deb11u1) para a Alta Disponibilidade. Para o Balanceamento de Carga, foi utilizado o software HAProxy (versão: 2.2.9-2+deb11u3).

Ao todo, o cluster é composto por seis nodos, cinco deles dedicados ao banco de dados, interligados em alta disponibilidade no modo master-master. Um nó dedicado ao balanceamento de carga, onde ele é responsável por receber as requisições e distribuir entre os nodos de acordo com seu algoritmo.

RESULTADOS

Este artigo nasceu como continuação do artigo “CLUSTER DE ALTA DISPONIBILIDADE COM BALANCEAMENTO DE CARGA EM MÁQUINAS VIRTUAIS: Gerenciando banco de dados MariaDB com Galera Cluster” com a proposta de tratar a escalabilidade do sistema clusterizado com softwares de virtualização.

O escalonamento horizontal foi aplicado aos clusters aumentando a sua quantidade de nodos de dois para cinco interligado com o banco de dados MariaDB e gerido pelo Galera Cluster. O processo foi feito através da clonagem das máquinas já existentes no artigo anterior a este, fazendo as devidas atualizações nos arquivos de configurações.

Os resultados da clonagem das máquinas foram satisfatórios, levando em consideração que em alguns momentos foi necessário intervenção manual para que a máquina entrasse em funcionamento. O Galera Cluster conseguiu fazer a gestão e entendeu as bases de dados que já adivinham das máquinas clonadas, evitando a resincronização dos dados. Após atualizadas as configurações do HA Proxy, o sistema entrou em funcionamento com a distribuição de requisições ao longo da estrutura.

A pesquisa esbarrou na limitação de hardware das máquinas, as duas máquinas utilizadas possuem quatro threads de processamento cada, então a pesquisa reservou um thread para máquina real e um thread para cada máquina virtual, de mesma forma as memórias, dos quatro gigabytes disponíveis, cada máquina, seja real ou virtual, recebeu um gigabyte para o sistema.

CONCLUSÃO

Este trabalho teve por objetivo continuar os estudos sobre a clusterização de serviços com máquina desktop instalando um cluster de balanceamento de carga com alta disponibilidade do banco de dados MariaDB com Galera Cluster em máquinas virtuais com Oracle VirtualBox. Para isso utilizou-se um computador desktop, com sistema Debian Linux instalado, dentro de uma rede interna da Faculdade de Tecnologia de Tatuí.

Nesta etapa o cluster já utilizado em outro artigo teve seus recursos aumentados com a escalabilidade horizontal do recurso, subindo de dois para cinco nodos interligados em um cluster de alta disponibilidade do tipo master-master, onde qualquer nó pode receber a requisição e replicar aos demais na estrutura. O balanceador de carga demonstrou eficiência em redirecionar as requisições para todos os servidores interligados. Assim este trabalho abre a possibilidade para uma continuidade dos trabalhos, visando a escalabilidade horizontal do sistema com adição de mais dispositivos.

Este trabalho foi concebido dentro da pesquisa de clusterização que o autor desenvolve dentro da Faculdade de Tecnologia de Tatuí-SP, Centro Paula Souza.

REFERÊNCIAS

About VirtualBox. **VirtualBox**, 2022. Disponível em: <<https://www.virtualbox.org/>>. Acesso em: 19 de set. 2022.

Como fazer escalonamento automático de grupos de nós. **Google Cloud**, 2022. Disponível em: <https://cloud.google.com/compute/docs/nodes/autoscaling-node-groups?hl=pt-br#scale_out_and_scale_in>. Acesso em 15 out. 2022.

Escalabilidade em Cloud Computing com previsibilidade: benefícios e características. **Escola Superior de Redes**, 2021. Disponível em: <<https://esr.rnp.br/computacao-em-nuvem/escalabilidade-em-cloud-computing-com-previsibilidade/>>. Acesso em: Acesso em 15 out. 2022.

Getting Started with MariaDB Galera Cluster. **MariaDB**, 2013. Disponível em: <<https://mariadb.com/kb/en/getting-started-with-mariadb-galera-cluster/>> . Acesso em: 03 de mar. 2022.

HAJDARBEGOVIC, M. Hypervisors: A Comprehensive Guide. **Virtasant**, 2020. Disponível em: <<https://www.virtasant.com/blog/hypervisors-a-comprehensive-guide>>. Acesso em: 03 mar. 2022.

O que é virtualização?. **RedHat**, 2018. Disponível em: <<https://www.redhat.com/pt-br/topics/virtualization/what-is-virtualization>> Acesso em: 01 ago. 2022.

PEREZ A. Sobre a História da Computação Distribuída e Clusters Kubernetes. **Data Team Stone**, 2021. Disponível em: <<https://medium.com/team-data-stone/sobre-a-hist%C3%B3ria-da-computa%C3%A7%C3%A3o-distribu%C3%ADa-e-clusters-kubernetes-3d0fe331db7>>. Acesso em: 25 de abr. 2022.

PITANGA, M. Computação em cluster. **Clube do Hardware**, 2003 Disponível em: <<https://www.clubedohardware.com.br/artigos/redes/computa%C3%A7%C3%A3o-em-cluster-r33711/>>. Acesso em: 01 de abr. 2022.

TELES, F. O que é Cluster e como essa estrutura pode ser benéfica para você. **Desk Manager**, 2018 Disponível em: <<https://deskmanager.com.br/blog/cluster/>>. Acesso em: 25 de abr. de 2022.