

LIGA DE ENSINO DO RIO GRANDE DO NORTE
CENTRO UNIVERSITÁRIO DO RIO GRANDE DO NORTE
ESPECIALIZAÇÃO EM REDES DE COMPUTADORES

MARCELO LEANDRO TAVARES MEDEIROS

ALTA DISPONIBILIDADE DE ARMAZENAMENTO COM O GLUSTERFS

NATAL/RN

2017

MARCELO LEANDRO TAVARES MEDEIROS

ALTA DISPONIBILIDADE DE ARMAZENAMENTO COM O GLUSTERFS

Trabalho de Conclusão de Curso apresentado ao Centro Universitário do Rio Grande do Norte (UNI-RN), como requisito final para obtenção do título de Especialista em Redes de Computadores.

Orientador: Prof. M.Sc. Aluizio Ferreira da Rocha Neto.

NATAL/RN

2017

Catálogo na Publicação – Biblioteca da UNI-RN
Setor de Processos Técnicos

Medeiros, Marcelo Leandro Tavares.

Alta disponibilidade de armazenamento com o GlusterFS / Marcelo
Leandro Tavares Medeiros. – Natal, 2017.

31 f.

Orientador: Prof. M.Sc. Alúzio Ferreira da Rocha Neto.

Monografia (Especialização em Redes de Computadores) – Centro
Universitário do Rio Grande do Norte.

1. Alta disponibilidade – Monografia. 2. Custo – Monografia. 3.
GlusterFS – Monografia. 4. Infraestrutura – Monografia. I. Rocha Neto,
Alúzio Ferreira da. II. Título.

RN/UNI-RN/BC

CDU 004.72

MARCELO LEANDRO TAVARES MEDEIROS

ALTA DISPONIBILIDADE DE ARMAZENAMENTO COM O GLUSTERFS

Trabalho de Conclusão de Curso apresentado ao Centro Universitário do Rio Grande do Norte (UNI-RN), como requisito final para obtenção do título de Especialista em Redes de Computadores.

Aprovado em: ____/____/____.

BANCA EXAMINADORA

Prof. M.Sc. Alúzio Ferreira da Rocha Neto
Orientador

Professor Convidado
Membro

Professor Convidado
Membro

Dedico este trabalho aos meus pais por eles sempre terem me incentivado, sempre me estimulando a seguir em frente, ficando ao meu lado nas minhas escolhas mesmo quando não concordava.

AGRADECIMENTO

Agradeço em primeiro lugar a Deus por ter me proporcionado mais essa conquista.

A minha família que me proporcionou toda a estrutura para a realização dessa pós-graduação.

Aos meus colegas de trabalho pelo auxílio na realização desse trabalho.

Ao professor Alúzio Ferreira da Rocha Neto, pelo auxílio na realização desse trabalho.

RESUMO

Com o grande crescimento da TI, as empresas estão gerando uma grande quantidade de dados, esses dados são de suma importância para o funcionamento das organizações. Devida a essa necessidade foram desenvolvidas várias formas de se armazenar esses dados. Para se montar um ambiente que comporte todos esses dados as empresas têm que fazer um investimento, seja locando espaço em ambientes remotos ou montando seu próprio ambiente. Para se montar uma infraestrutura de alta confiabilidade é custoso pois precisamos investir em diversos equipamentos, tanto na parte lógica como na parte estrutural. Neste trabalho utilizaremos o GlusterFS, que fornece um ambiente de alta disponibilidade de armazenamento reduzindo o custo que as ferramentas convencionais oferecem.

Palavras-chave: Alta disponibilidade. Custo. GlusterFS. Infraestrutura.

ABSTRACT

With the great growth of IT, companies are generating a lot of data, these data are of paramount importance to the functioning of organizations. Due to this need several ways of storing these data have been developed. In order to set up an environment that supports all these data companies have to make an investment, either by renting space in remote environments or by setting up their own environment. To build a high reliability infrastructure is expensive because we need to invest in various equipment, both in the logical and in the structural part. In this work we will use GlusterFS, which provides a high availability storage environment, reducing the cost that conventional tools offer

Keywords: High availability. Cost. GlusterFS. Infrastructure.

LISTA DE FIGURAS

Figura 1 – Arquitetura do GlusterFS	15
Figura 2 – Volume Striped	17
Figura 3 – Volume Distributed	18
Figura 4 – Volume Replicated	19
Figura 5 – Volume Distributed Replicated	20
Figura 6 – Volume Distributed Striped	21
Figura 7 – Arquitetura do laboratório	22
Figura 8 – Verificando montagem do diretório	24
Figura 9 – Verificando montagem do diretório no cliente	27
Figura 10 – Criando arquivo teste01	28
Figura 11 – Listando arquivos do brick no servidor01	28
Figura 12 – Listando arquivos do brick no servidor02	28
Figura 13 – Verificando status do servidor01	29
Figura 14 – Criando arquivo teste02	29
Figura 15 – Listando arquivos do brick no servidor01	29

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
Ext3	Third Extended Filesystem
FAT32	File Allocation Table
GB	Gigabyte
Hot Spare	Disco de Segurança
IP	Internet Protocol
NFS	Network File System
NTFS	New Technology File System
RAID	Redundant Array of Independent Disks
SMB	Server Message Block
TCP	Transmission Control Protocol
XFS	X-Windows File Server

SUMÁRIO

1 INTRODUÇÃO	11
1.1 OBJETIVOS	11
1.1.1 Geral	11
1.1.2 Específicos	12
2 REFERENCIAL TEÓRICO	13
2.1 STORAGE.....	13
2.2 SISTEMA DE ARQUIVO	13
2.3 REDUNDÂNCIA	13
2.4 INTEGRIDADE.....	14
2.5 GLUSTERFS.....	14
3 DESENVOLVIMENTO DA SOLUÇÃO	15
3.1 FUNCIONAMENTO DO GLUSTERFS.....	15
3.2 CONCEITOS DO GLUSTERFS	16
3.2.1 Brick	16
3.2.2 Cliente	16
3.2.3 Server	16
3.2.4 Volume	16
3.3 CONFIGURAÇÃO DO GLUSTERFS	16
3.3.1 Striped	17
3.3.2 Distributed	17
3.3.3 Replicated	18
3.3.4 Distributed Replicated	19
3.3.5 Distributed Striped	20
3.4 LABORATÓRIO	21
3.5 CONFIRANDO O SERVIDOR GLUSTERFS	22
3.5.1 Instalando GlusterFS nos servidores Centos 7	22
3.5.2 Preparando os HDs	22
3.5.3 Criar os bricks	23
3.5.4 Serviço de nomes.....	24
3.5.5 Liberação no Firewall.....	25
3.5.6 Iniciando o serviço GlusterFS.....	25
3.5.7 Configurando serviço GlusterFS	25

3.5.7.1 Criando o pool de servidores.....	25
3.5.7.2 Criando o volume	26
3.6 CONFIGURANDO NO CLIENTE CENTOS 7.....	26
3.7 UTILIZAÇÃO	27
4 CONCLUSÃO	30
REFERÊNCIAS.....	31

1 INTRODUÇÃO

Atualmente é praticamente impossível para uma empresa se manter sem acesso à tecnologia, seja uma infraestrutura local, uma página web ou uma aplicação em nuvem. É crucial para qualquer empresa que se preze obter informação sobre o que seus clientes ou sobre o mercado que ela pertence, e conseqüentemente é gerado um grande volume de informações que precisam ser bem guardadas e armazenadas em um local seguro e confiável para uma análise posterior.

Para atender a essa demanda existem diversas ferramentas no mercado, cada uma com sua proposta e finalidade, cabendo a empresa escolher a que mais se encaixa as suas necessidades. Uma das necessidades mais buscadas é a relação baixo custo e alta confiabilidade. O GlusterFS é uma ferramenta de sistema de arquivo distribuído que se encaixa muito bem a essa necessidade por ser open source, ou seja, sem custos, e de alta confiabilidade. Para configuração não é necessário nenhum tipo de hardware específico, sendo possível montar com servidores de baixo custo, sem a necessidade de controladora que é a parte mais custosa de um storage, além da diminuição do custo com hardware, por se tratar de servidores menos robustos há uma diminuição do consumo de energia, reduzindo assim os gastos pra se montar um ambiente de alta confiabilidade.

Para esse trabalho foi montado um laboratório utilizando maquinas virtuais onde o GlusterFS será instalado em dois servidores que utilizam o CentOS 7 como sistema operacional, também será configurado um cliente com CentOS 7 para realização de testes.

1.1 OBJETIVOS

1.1.1 Geral

O objetivo desse trabalho é analisar e conhecer a ferramenta GlusterFS, realizando sua configuração em um ambiente de teste para analisar seu comportamento, verificar seu desempenho, redundância e integridade, além do seu comportamento ao simular uma falha de hardware.

1.1.2 Específicos

Para atingir o objetivo geral, os seguintes objetivos foram definidos para melhor avaliar a ferramenta GlusterFS em um ambiente real:

- Reduzir o custo na montagem da infraestrutura
- Fornecer um serviço com um melhor desempenho
- Oferecer maior confiabilidade
- Fornecer um serviço com redundância
- Realizar teste de integridade e disponibilidade

2 REFERENCIAL TEÓRICO

2.1 STORAGE

Storages são dispositivos projetados especificamente para armazenamento de dados, onde através de uma conexão via rede, você pode conectar seu(s) servidor(es) à um storage, facilitando assim a expansão da capacidade de armazenamento sem impacto na produção, garantindo maior flexibilidade e confiabilidade no armazenamento (LOCAWEB, 2017).

Esses dispositivos em sua maioria possuem controladoras que são responsáveis por gerenciar os discos, essas controladoras possuem uma função que agrupam os discos, essa função se chama RAID, esse agrupamento permite que o Storage oferece um grande volume formado pela junção dos discos agrupados, de acordo com as melhores práticas, deve-se sempre deixar um disco de reserva (hot spare), para que caso ocorra problema em algum disco de produção o de reserva assumirá seu lugar, garantido assim a continuidade do serviço.

2.2 SISTEMA DE ARQUIVO

Um sistema de arquivos é um conjunto de estruturas lógicas e de rotinas, que permitem ao sistema operacional controlar o acesso ao disco rígido. Diferentes sistemas operacionais usam diferentes sistemas de arquivos (MORIMOTO, 2017).

Os principais sistemas de arquivos utilizados pelo Windows são NTFS e FAT32, já o Linux utiliza sistemas diferentes tais como EXT4, XFS e etc. Cada sistema de arquivo possui suas particularidades, alguns permitem a expansão e diminuição de tamanho de suas partições, outros apenas a expansão, também há limites quando ao tamanho máximo de um arquivo que cada sistema de arquivo consegue reconhecer. Dessa forma cabe ao usuário decidir qual é o mais indicado de acordo com sua necessidade.

2.3 REDUNDÂNCIA

"Redundância é o meio mais simples de obter um sistema de alta disponibilidade" (WIKIPÉDIA, 2017).

Redundância é uma forma de garantir a continuidade do serviço caso algum

equipamento que compõem esse serviço venha a apresentar falhas. Existem diversos tipos de redundância como de placa de rede, CPU, servidores entre outros, a redundância que será tratada nesse trabalho é a de disco e servidor.

2.4 INTEGRIDADE

[...] propriedade que garante que a informação manipulada mantenha todas as características originais estabelecidas pelo proprietário da informação, incluindo controle de mudanças e garantia do seu ciclo de vida (nascimento, manutenção e destruição) (REDAÇÃO..., 2017).

Termo utilizado em situações que garantem que algo esteja ou continue da mesma forma que foi criado mesmo após o envio para um local diferente da sua origem. Na área de informática este termo é muito utilizado quando um arquivo é enviado de um dispositivo para outro e mantém todas as suas características originais.

2.5 GLUSTERFS

GlusterFS is an open source, distributed file system capable of scaling to several petabytes and handling thousands of clients. It is a file system with a modular, stackable design, and a unique no-metadata server architecture (GLUSTERFS, 2017).

Conforme informado pelo fabricante o GlusterFS é um software open source de sistemas distribuídos capaz de escalar e manipular milhares de clientes. Possui um design empilhável fornecendo assim uma maior escalabilidade. Possui uma arquitetura sem metadados o que garanti um melhor desempenho e confiabilidade. O GlusterFS foi desenvolvido pela Red Hat, porém, por ser *open source* é mantido por vários contribuintes. *Atualmente* ele é utilizado por milhares de empresas que abrangem os setores de mídia, saúde, governo, educação e etc.

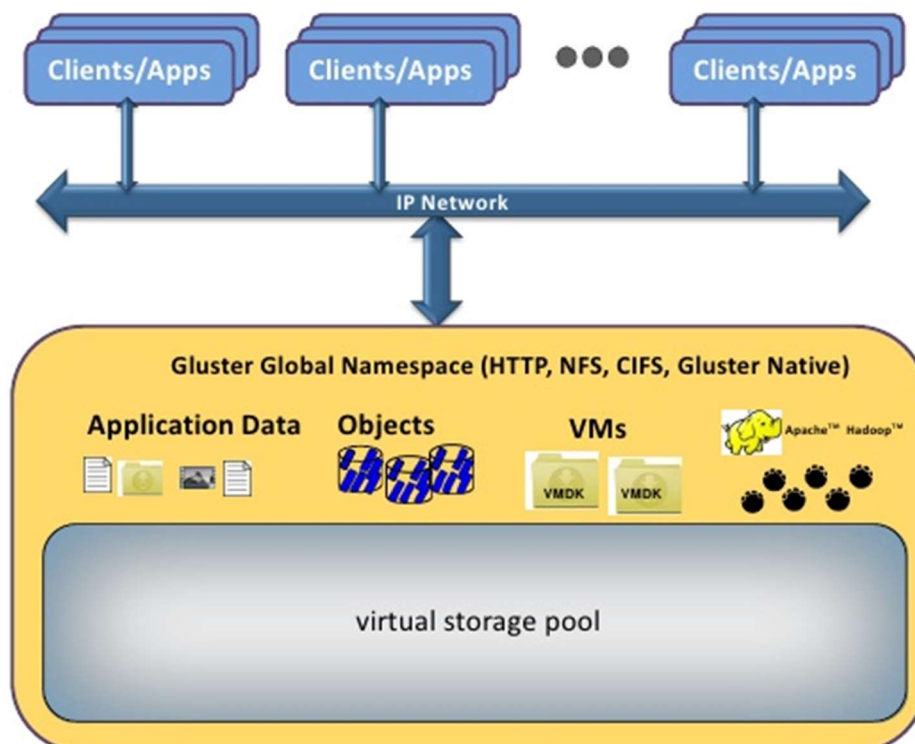
3 DESENVOLVIMENTO DA SOLUÇÃO

3.1 FUNCIONAMENTO DO GLUSTERFS

O GlusterFS pode ser utilizado com recursos físicos ou virtuais, oferecendo um armazenamento altamente disponível e eficiente a um baixo custo. O GlusterFS agrupa os blocos de *storage* sobre TCP/IP, agregando recursos de discos memória e gerenciando dados em um único *namespace* global.

O GlusterFS agrega vários servidores de armazenamento através de interconexões de rede em um grande sistema de arquivos. Baseado em um espaço empilhável ele fornece um desempenho excepcional para diferentes tipos de utilização. Os servidores GlusterFS usam qualquer tipo de sistema de arquivos atributos estendidos (ext4, XFS, etc.), os dados podem ser acessados utilizando os principais protocolos de acesso via rede (NFS e SMB). O GlusterFS suporta qualquer tipo de cliente utilizando qualquer aplicativo como mostra a Figura 1.

Figura 1 – Arquitetura do GlusterFS



Fonte: GlusterFS (2017).

3.2 CONCEITOS DO GLUSTERFS

3.2.1 Brick

É o nome dado ao diretório raiz onde GlusterFS monta sua estrutura de volumes, em uma simples analogia, cada *brick* funciona como um diretório dentro de uma partição, por exemplo pode-se criar 2 *bricks* como */brick1 /brick2* que seria semelhante a */home* ou */var*. Seguindo as melhores práticas cada *brick* deve ser configurado em um disco, e um disco não pode contar mais de um *brick*.

3.2.2 Cliente

Dispositivo da rede que irá se conectar ao volume e irá utilizar os serviços de armazenamento. O GlusterFS possui uma API que faz com que a unidade montada pareça um disco local.

3.2.3 Server

Maquina (virtual ou física) onde será instalado o GlusterFS e que hospeda o sistema de arquivos no qual os dados serão armazenados.

3.2.4 Volume

Diretório compartilhado que será utilizado pelo cliente, este volume é formado pelos *bricks*.

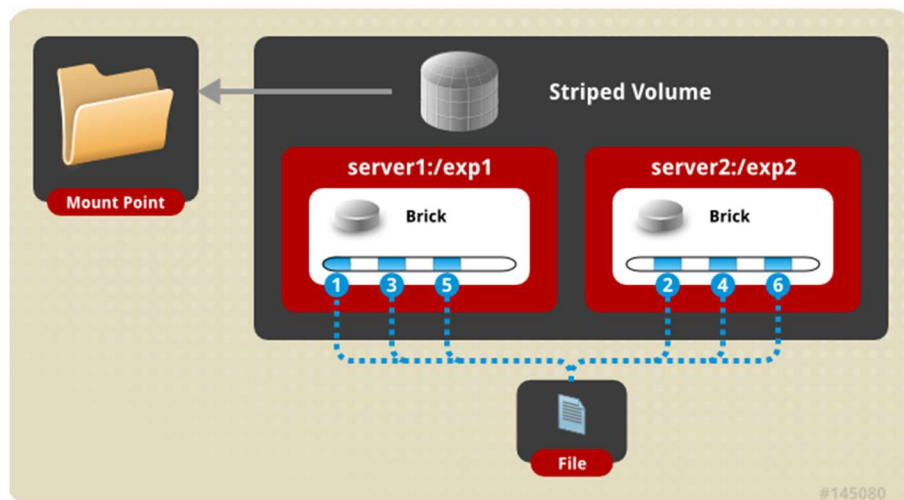
3.3 CONFIGURAÇÃO DO GLUSTERFS

O GlusterFS tem 3 formas básicas de configuração *striped*, *replicated* e *distributed*. Essas três configurações podem trabalhar juntas obtendo assim melhores resultados e redundância.

3.3.1 Striped

O volume *striped* divide os dados entre os *bricks*. Isso significa que cada arquivo é dividido em várias partes e armazenados em *bricks* diferentes. O *striped* é uma das configurações menos utilizando pela falta de redundância, se algum *brick* apresenta problema a informação será perdida, com isso ele se torne inviável para ambiente de alta criticidade, a sua vantagem é que o arquivo está em vários *bricks* com isso ele utiliza o desempenho de todos, o que o deixa mais rápido.

Figura 2 – Volume Striped



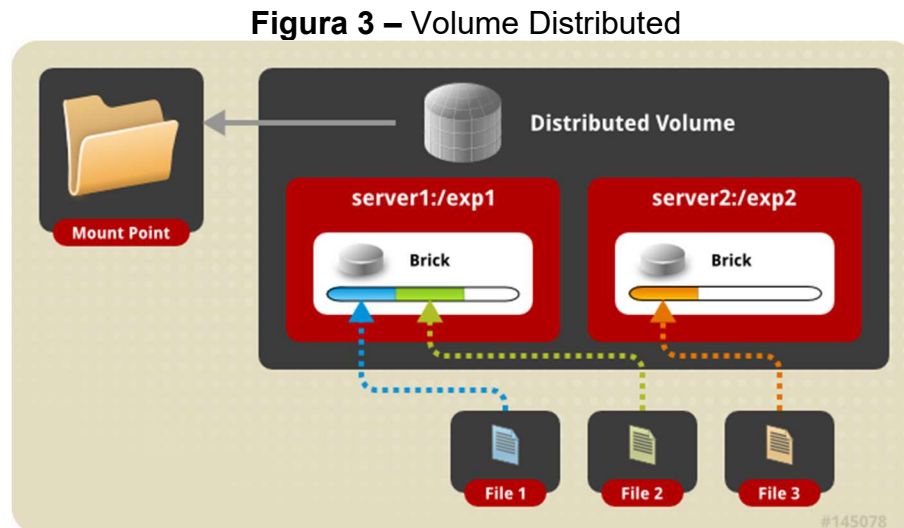
Fonte: GlusterFS (2017).

Como observado na figura2, server1 e server2 são os servidores com o GlusterFS, foi configurando um *brick* em cada servidor, com isso temos 2 *bricks*, utilizando o volume *striped*, o espaço dos *bricks* são somados e entregues como apenas um volume, com isso se tivermos 2 *bricks* cada um com 10GB, no final teremos 20GB. Percebe-se na imagem também que o arquivo é dividido em várias partes e essas partes são distribuídas entre os *bricks*.

3.3.2 Distributed

Este é o volume padrão utilizando pelo GlusterFS, se na hora da criação não especificarmos o tipo do volume este será adotado. Os arquivos são armazenados de forma aleatória entre os *bricks*. Sendo assim um arquivo só pode estar armazenado em um *brick*, portanto não a redundância de dados, esse volume é

mais utilizado para ambientes de baixa criticidade e necessita de um grande volume de espaço. Muito cuidado pois a perda de um *brick* acarretará na perda dos dados nele contido.



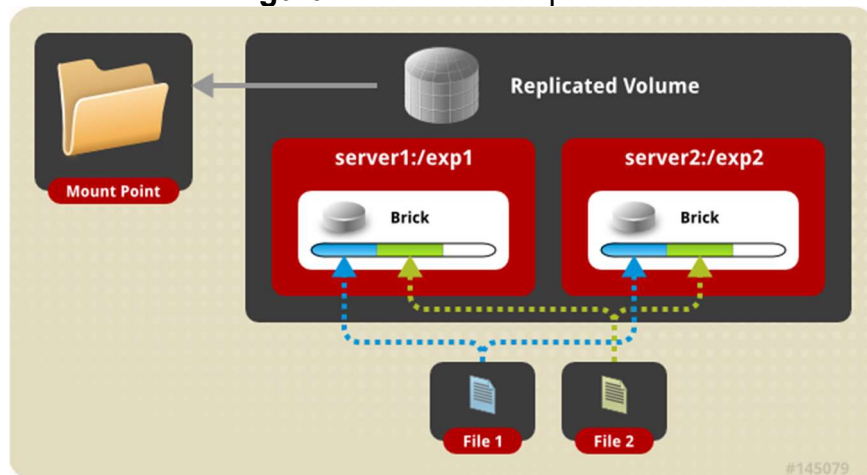
Fonte: GlusterFS (2017).

Como observado na figura2, server1 e server2 são os servidores com o GlusterFS, foi configurado um *brick* em cada servidor, com isso temos 2 *bricks*, utilizando o volume *distributed*, os espaços dos *bricks* são somados, com isso se tivermos 2 *bricks* cada um com 10 GB, no final temos um volume de 20GB. Percebe-se que os arquivos são distribuídos entre os *bricks*, cada arquivo ficando em apenas um *brick*, sendo assim se tivermos uma falha de *brick*, as informações nele contida será perdida, já os arquivos que estão no outro *brick* ficará acessível.

3.3.3 Replicated

Esse volume é utilizado para garantir um ambiente redundante, isso significa que o mesmo arquivo é salvo em *bricks* diferentes, portanto se um apresenta problema a informação ainda está disponível. No momento da criação do volume o administrador define quantas replicações do arquivo iram existir, sendo assim se quisermos uma réplica de 2 precisamos de no mínimo 2 *bricks*, e se quisermos replica de 3 precisamos de pelo menos 3 *bricks*.

Figura 4 – Volume Replicated



Fonte: GlusterFS (2017).

Como observado na figura4, server1 e server2 são os servidores com o GlusterFS, foi configurando um *brick* em cada servidor, com isso temos 2 *bricks*, utilizando o volume *replicated*, os espaços dos *brick* não são somando tendo em vista que as mesmas informações contidas em um ficaram nos demais, sendo assim se tiver 2 *bricks* cada um com 10GB, ao final teremos 10GB de armazenamento para o usuário. Observa-se que o mesmo arquivo é salvo em ambos os *bricks*, caso ocorra a falha de algum *brick* as informações ainda estarão acessíveis.

3.3.4 Distributed Replicated

Neste volume os arquivos são distribuídos entre conjuntos replicados de bricks, o número de *brick* deve ser um múltiplo da contagem de replica. A ordem que especificamos na criação do *brick* será obedecida, se adicionarmos na ordem brick1 brick3 brick2 brick4, o brick1 será redundante com o brick3 e assim por diante, esse tipo de volume é utilizado quando o ambiente necessita de alta disponibilidade e de espaço, pois ele faz a redundância de *brick* e junta as redundâncias formando um grande volume distribuído e replicado.

Formula para saber quantos *bricks* são necessários:

Numero de *bricks* ÷ Numero de replica = Número inteiro

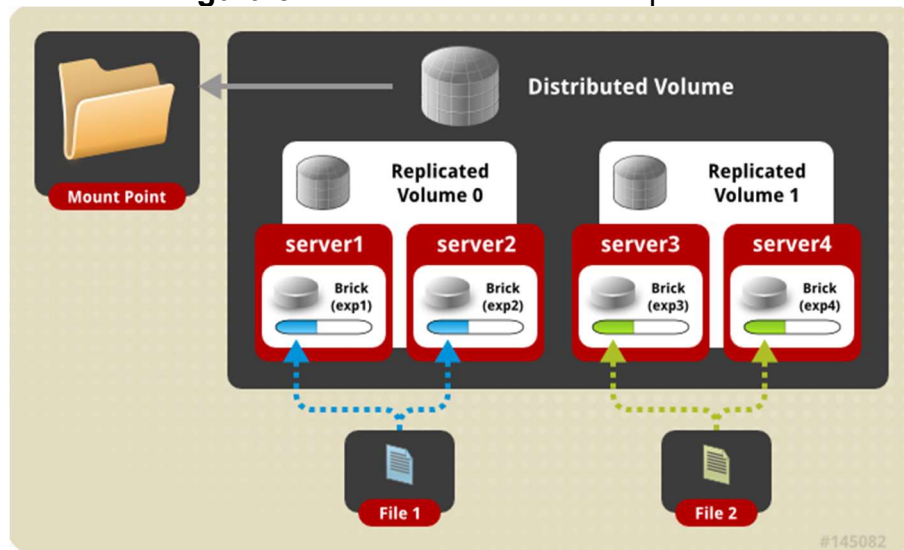
$$6 \div 2 = 3$$

Se tiver 6 *bricks* pode-se utilizar a replica 2 pois 6 é múltiplo de 2.

$$6 \div 4 = 1,5$$

Se tiver 6 *bricks* não pode utilizar replica 4 pois 6 não é múltiplo de 4.

Figura 5 – Volume Distributed Replicated



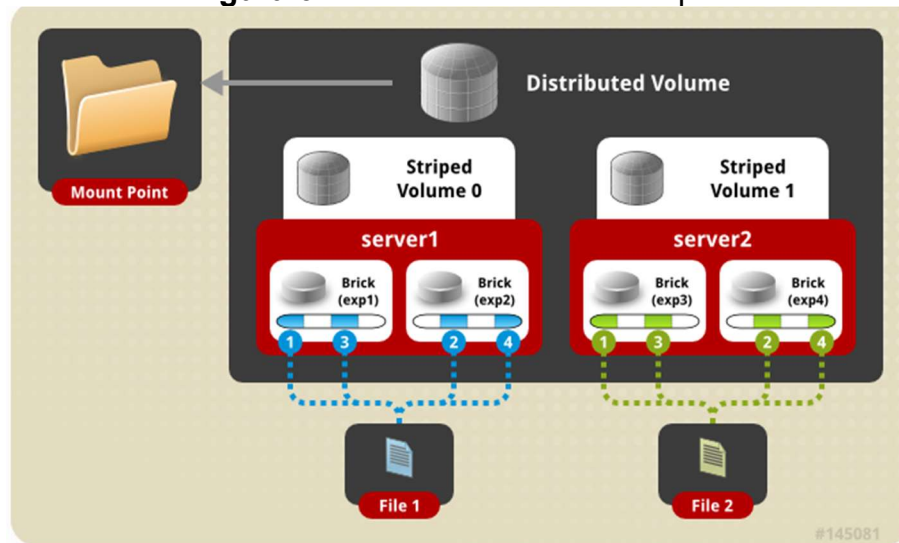
Fonte: GlusterFS (2017).

Como observado na figura05, server1, server2, server3 e server4 são os servidores com o GlusterFS, foi configurado um *brick* em cada servidor, com isso temos 4 *bricks*, utilizando o volume *distributed replicated*, teremos 2 volumes replicados, volume0 que é formado pelo server1 e server2 e volume1 que é por servidor3 e servidor4, ao final temos um volume formado pelos 2 replicado, junto todo o espaço do volume 0 e 1, criando assim um grande e único volume juntando o espaço dos volumes replicados. Como observa-se os arquivos são salvos separadamente por volume e replicados nos *bricks* que compõem esses volumes.

3.3.5 Distributed Striped

Este volume é semelhante ao *striped*, porém nesse modo os arquivos podem ser distribuídos através de mais quantidades de *bricks*. O volume de *bricks* deve ser um múltiplo do número de *stripeds*, com isso, se quisermos aumentar o tamanho, deve-se adicionar *bricks* múltiplos da contagem de *stripeds*.

Figura 6 – Volume Distributed Striped



Fonte: GlusterFS (2017).

Como observado na figura6, server1 e server2 são os servidores com o GlusterFS, foram configurados dois *bricks* em cada servidor, com isso temos 4 *bricks*, utilizando o volume *Distributed Striped*, teremos 2 volumes *striped*, volume0 formado pelo server1 e volume1 formado pelo server2, cada servidor contém 2 *bricks*, sendo o arquivo compartilhado entre os bricks do mesmo servidor. Ambos os volumes são somados formando um volume distribuído.

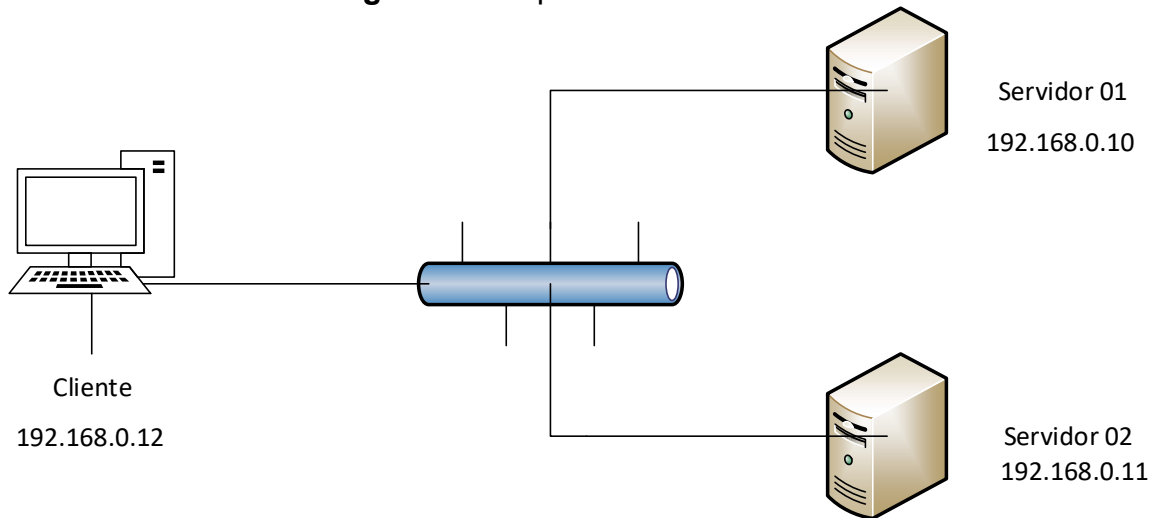
3.4 LABORATÓRIO

Para esse projeto foi realizado um laboratório que contém dois servidores que serão utilizados como GlusterFS, esses servidores contarão com 1 processador, 512mb, dois discos de 10GB e uma placa de rede, O sistema operacional será o Centos 7. Como cliente será utilizando uma máquina com Centos 7, o cliente irá acessar o volume via protocolo do próprio GlusterFS. O volume utilizado pelo GlusterFS será o replicado pois o mesmo é o melhor custo benefício e já fornece a segurança necessária para um ambiente de alta criticidade que não necessita de um ótimo desempenho.

O foco principal do laboratório será demonstrar a instalação e configuração do GlusterFS assim como também a configuração do cliente, não fazendo parte do escopo do laboratório a instalação dos sistemas operacionais que serão utilizados no servidor, Centos 7, e da máquina cliente, Centos 7. Para este laboratório está

sendo levando em consideração que o sistema operacional já foi instalado.

Figura 7 – Arquitetura do laboratório



Fonte: Elaborado própria, 2017.

A figura 07 ilustra como será a estrutura do laboratório, onde os servidores 1 e 2 serão utilizados como servidores GlusterFS e onde ficaram configurados os bricks. O Cliente irá se conectar aos servidores pela rede utilizando o cliente do GlusterFS. Para esse laboratório utilizaremos os IPs na faixa de rede 192.168.0/24.

3.5 CONFIRANDO O SERVIDOR GLUSTERFS

3.5.1 Instalando GlusterFS nos servidores Centos 7

Para instalar o software do GlusterFS é necessário executar os seguintes comandos em ambos os servidores, servidor01 e servidor02. O '#' antes do comando informa que é necessário executá-los como root do sistema.

```
#yum install centos-release-gluster310
#yum install glusterfs-server
```

3.5.2 Preparando os HDs

O primeiro passo é particionar o HD, o particionamento pode ser de duas formas, dividir os discos em várias partições ou em apenas uma, o indicado para

esse tipo de ferramenta é utilizar apenas uma partição, pois será nessa partição que o diretório do *brick* será criado, e segundo as boas práticas dos desenvolvedores, cada *brick* deve ter seu próprio HD, todos os passos a baixo devem ser executados em ambos os servidores, servidor01 e servidor02. Uma das formas de se particionar disco é utilizando “fdisk”, segue o passo a passo.

```
#fdisk /dev/vdb
```

Onde vdb é o nome dado para o disco pelo sistema operacional, para verificar o nome do disco, utilize o comando `fdisk -l`, esse comando lista todos os HDs do sistema.

Após executar o comando anterior a tela do “fdisk /dev/vdb” será exibida uma nova tela e deve-se seguir os passos detalhados a frente para executar o particionamento completo do HD.

Na primeira opção deve-se digitar "n" e apertar "enter", essa opção informa que será criada uma nova partição, nas demais opções deve-se selecionar as opções padrões, apenas apertando "enter", ao final para sair do fdisk digitar "w" salvando as alterações realizadas no HD.

Após a criação da partição é necessário configurar o sistema de arquivo que será utilizada pela partição, por recomendação dos desenvolvedores do GlusterFS foi utilizado o sistema de arquivo XFS, o seguinte comando deve ser executado em ambos os servidores, servidor01 e servidor02.

```
#mkfs.xfs /dev/vdb1
```

Para cada partição criada o Linux adiciona um número ao nome do HD para informa qual a partição, como só temos uma partição executamos apenas nela.

3.5.3 Criar os bricks

Com os HDs particionados é hora de criar os *bricks*, esse comando deve ser executado em ambos os servidores, servidor01 e servidor02.

```
#mkdir -p /data/brick1
```

Obs: Como informado anteriormente os *bricks* são diretórios e podem ser criados em qualquer local, foi adicionado ao comando o parâmetro "-p" para que seja possível criar subpastas.

O comando a seguir automatizar a montagem da partição criada anteriormente no diretório que será utilizado como *brick*, com isso os dados adicionados no diretório serão salvos nesse sistema de arquivo, esses comandos devem ser executados em ambos os servidores, servidor01 e servidor02.

```
#echo '/dev/vdb1 /data/brick1 xfs defaults 1 2' >> /etc/fstab
#mount -a && mount
```

Após a execução é possível ver que o diretório foi montado na partição vdb1 com o comando `df -h`.

Figura 8 – Verificando montagem do diretório

```
[root@servidor01 ~]# df -h
Sist. Arq.      Tam. Usado Disp. Uso% Montado em
/dev/mapper/cl-root 7,1G  1,3G  5,9G  18% /
devtmpfs        234M    0  234M   0% /dev
tmpfs           245M    0  245M   0% /dev/shm
tmpfs           245M  4,4M  240M   2% /run
tmpfs           245M    0  245M   0% /sys/fs/cgroup
/dev/vda1       1014M  184M  831M  19% /boot
/dev/vdb1       10G    33M   10G   1% /data/brick1
tmpfs           49M    0   49M   0% /run/user/0
```

Fonte: Elaborado própria, 2017.

3.5.4 Serviço de nomes

O GlusterFS utiliza o serviço de nomes para realizar suas conexões, se não possuir um servidor de nomes, adicione as seguintes entrada no arquivo `/etc/hosts` utilizando o editor de texto a sua escolha. Adicionar as entradas em ambos os servidores, servidor01 e servidor02.

```
192.168.0.10 servidor01 # ip e nome do servidor01
```

```
192.168.0.11 servidor02 # ip e nome do servidor02
```

3.5.5 Liberação no Firewall

É necessário adicionar uma liberação no firewall para que os servidores se comuniquem, como se trata de um laboratório libramos toda a rede dos servidores. Executar os comandos abaixo em ambos os servidores, servidor01 e servidor02.

3.5.6 Iniciando o serviço GlusterFS

```
#firewall-cmd --permanent --zone=trusted --add-source=192.168.0.0/24  
#firewall-cmd --reload
```

Após preparar o sistema iniciar o serviço do GlusterFS com o seguinte comando, esse comando deve ser executado em ambos os servidores, servidor01 e servidor02:

```
#service glusterd start
```

Para verificar o status do serviço utilizar o comando:

```
#service glusterd status
```

```
#systemctl enable glusterd
```

Configurar para que o serviço do GlusterFS inicie junto com o sistema.

3.5.7 Configurando serviço GlusterFS

3.5.7.1 Criando o pool de servidores

Para criar o pool de servidores utiliza-se o seguinte comando no servidor01.

```
#gluster peer probe servidor02
```

Este comando pode ser executado em ambos os servidores, substituindo o nome do servidor para o qual desejamos conectar. Após a criação do pool apenas

servidores que pertencem ao pool podem adicionar novos servidores.

3.5.7.2 Criando o volume

Com o pool criado é hora de criar o volume, como informado anteriormente o volume utilizado será o replicado, executar o seguinte comando no servidor01 ou no servidor02.

```
#gluster volume create gv0 replica 2 servidor01:/data/brick1/gv0  
servidor02:/data/brick1/gv0 force
```

No comando acima o nome "gv0" representa o nome do volume, "replica" o tipo do volume. "2" a quantidade de réplica e os demais o nome e o diretório brick de cada servidor.

Após criar o volume é necessário iniciar o mesmo, executar o seguinte comando no servidor01 ou servidor02.

```
#gluster volume start gv0
```

O volume está pronto, porem para ser utilizado pelos clientes são necessárias algumas configurações extras. Para utilizar com o Windows é necessário configurar um servidor NFS, já para clientes Linux pode-se instalar o próprio cliente do GlusterFS, esse cliente não está disponível para Windows.

3.6 CONFIGURANDO NO CLIENTE CENTOS 7

A comunicação do cliente com o servidor utilizara o GlusterFS-client, a instalação é bem simples, basta utilizar o comando abaixo.

```
#yum install glusterfs-client
```

Como informado anteriormente o GlusterFS utiliza o serviço de nomes para realizar suas conexões, se não possuir um servidor de nomes, adicione as seguintes entrada no arquivo /etc/hosts utilizando o editor de texto a sua escolha.

192.168.0.10 servidor01 # ip e nome do servidor01

192.168.0.11 servidor02 # ip e nome do servidor02

Após a instalação será necessário apenas montar o volume do GlusterFS na pasta que desejar com o seguinte comando.

```
#mount -t glusterfs 192.168.:/gv0 /mnt
```

Será utilizado o servidor01 para conectar a primeira vez no volume, ao se conectar ao volume o cliente identifica que tem outro servidor ligado ao volume e inicia a conexão automaticamente, o volume foi montado no diretório “/mnt” padrão de montagem do Linux, porém pode-se montar o volume em qualquer diretório.

Para confirma que o volume foi montado deve-se utilizar o comando `df -h` como ilustra a figura abaixo.

Figura 9 – Verificando montagem do diretório no cliente

Sist. Arq.	Tam.	Usado	Disp.	Uso%	Montado em
/dev/mapper/cl-root	7,1G	1,3G	5,9G	18%	/
devtmpfs	234M	0	234M	0%	/dev
tmpfs	245M	0	245M	0%	/dev/shm
tmpfs	245M	4,4M	240M	2%	/run
tmpfs	245M	0	245M	0%	/sys/fs/cgroup
/dev/vda1	1014M	166M	849M	17%	/boot
tmpfs	49M	0	49M	0%	/run/user/0
servidor01:/gv0	10G	33M	10G	1%	/mnt

Fonte: Elaborado própria, 2017.

Pronto, o volume está montado e pode ser utilizado por qualquer tipo de aplicação.

3.7 UTILIZAÇÃO

Para o laboratório será utilizado o volume replicado, como informado anteriormente esse volume garante a alta disponibilidade caso ocorra alguma falha em um dos servidores, para garantir esse serviço o GlusterFS duplica o arquivo, sendo armazenado uma cópia em casa brick. Para validar essa informação será criado um arquivo no volume e deverá ser verificar se a informação está armazenada em ambos os bricks.

No cliente que o volume foi montado ,deve ser criado um arquivo com o

comando touch, após a criação do arquivo verificar se o mesmo foi criado como ilustra a figura 10:

Figura 10 – Criando arquivo teste01

```
[root@cliente01 ~]# touch /mnt/teste01
[root@cliente01 ~]# ls -l /mnt/
total 0
-rw-r--r--. 1 root root 0 Abr  5 10:23 teste01
```

Fonte: Elaborado própria, 2017.

Verificar se o arquivo foi criado em ambos os bricks.

Acessando os servidores deve-se utilizar o comando ls -l para listar os arquivos dentro do brick, como será verificado nos dois servidores, esse comando deve ser executado tanto no servidor01 como no servidor02 conforme ilustrados na figura 11 e figura12:

Figura 11 – Listando arquivos do brick no servidor01

```
[root@servidor01 ~]# ls -l /data/brick1/volume01/
total 0
-rw-r--r--. 2 root root 0 Abr  5 10:23 teste01
```

Fonte: Elaborado própria, 2017.

Figura 12 – Listando arquivos do brick no servidor02

```
[root@servidor02 ~]# ls -l /data/brick1/volume01/
total 0
-rw-r--r--. 2 root root 0 Abr  5 10:23 teste01
```

Fonte: Elaborado própria, 2017.

Como observa-se nas imagens os arquivos foram adicionados em ambos os bricks.

Agora será simulado uma falha de servidor GlusterFS desligado o servidor01 e verificando se o arquivo ainda está acessível.

Após o desligamento do servidor01, deve-se utilizar o comando “#gluster peer status” no servidor02 para verificar o status dos servidores que compõem o pool, conforme ilustra a figura 13 o servidor01 está desconectado.

Figura 13 – Verificando status do servidor01

```
[root@servidor02 ~]# gluster peer status
Number of Peers: 1

Hostname: servidor01
Uuid: 93183782-faaa-40d2-83e4-d58ab43facac
State: Peer in Cluster (Disconnected)
[root@servidor02 ~]#
```

Fonte: Elaborado própria, 2017.

No cliente, deve-se criar um arquivo no volume montado para verificar se o acesso ao mesmo ainda está disponível.

Figura 14 – Criando arquivo teste02

```
[root@cliente01 ~]# touch /mnt/teste02
[root@cliente01 ~]# ls -l /mnt/
total 1
-rw-r--r--. 1 root root 8 Mar 30 16:09 teste
-rw-r--r--. 1 root root 0 Abr  5 09:43 teste02
```

Fonte: Elaborado própria, 2017.

Como nota-se, mesmo com servidor desligado o volume ainda se encontra disponível.

Ao restabelecer o servidor01, o servidor02 repassa todas os dados criados enquanto o mesmo estava indisponível, como ilustra a figura 16.

Figura 15 – Listando arquivos do brick no servidor01

```
[root@servidor01 ~]# ls -l /data/brick1/volume01/
total 4
-rw-r--r--. 2 root root 8 Mar 30 16:09 teste
-rw-r--r--. 2 root root 0 Abr  5 09:43 teste02
```

Fonte: Elaborado própria, 2017.

Com este último teste, o laboratório foi concluído.

4 CONCLUSÃO

Com implantação do GlusterFS, pode-se perceber que se trata de uma ferramenta de alta disponibilidade e de fácil configuração. O GlusterFS permite montar um ambiente de alta disponibilidade e oferecer um serviço de excelência, garantido a integridade e disponibilidade dos dados nele armazenados, possuindo diversas configurações o GlusterFS se adequa a qualquer ambiente, podendo ser configurado para fornecer velocidade, confiança ou um mix dos dois. Por se tratar de uma ferramenta open source e não possuir nenhuma restrição de hardware, a solução se torna bem acessível, pois pode ser implementada com hardware de baixo custo e mesmo assim garantir um ambiente robusto. Grandes empresas como HP e Lenovo fornecem uma solução já completa, redundante e com um suporte de alto nível, porém o custo para se montar um ambiente é elevado, em média 60 mil reais para um ambiente de médio porte, em contra partida o GlusterFS por não oferecer uma limitação de hardware pode ser montado em servidores Supermicro , que possuem todas as redundâncias necessárias e são mais acessíveis financeiramente, dando assim a possibilidade de obter um ambiente seguro ao custo bem mais reduzido, já o suporte pode ser obtido junto a Red Hat por um valor acessível. O GlusterFS pode ser utilizado em diversos cenários, por exemplo: se temos uma empresa que compartilha conteúdo em vários locais do mundo, ela pode utilizar o volume striped dividir essa informação entre vários bricks, e quando o cliente for acessar os dados o programa ira retorna apenas o que lhe interessa, não sendo necessário ler o arquivo por completo. Em ambientes que necessitam de desempenho como um data center, pode-se utilizar o volume distributed replicated, que fornece um ambiente com ótimo desempenho e totalmente tolerante a falhas, como os dados são distribuídos entre os servidores, o GlusterFS utiliza do processamento, memoria e disco de vários servidores para fornecer o acesso às informações, o que o torna mais rápido. Pode-se perceber que o GlusterFS se adequa a qualquer tipo de ambiente, cabendo assim o usuário decidir de qual forma irá utiliza-lo.

REFERÊNCIAS

GLUSTERFS. **Administration Guide**. Disponível em: <<http://gluster.readthedocs.io/en/latest/Administrator%20Guide/>>. Acesso em: 26 abr. 2017

LOCAWEB. **O que é um Storage?**. Disponível em: <https://wiki.locaweb.com.br/pt-br/Storage_-_O_que_%C3%A9_e_como_obter>. Acesso em: 24 abr. 2017.

MORIMOTO, Carlos. **Sistema de arquivos**. Disponível em: <<http://www.hardware.com.br/termos/sistema-de-arquivos>>. Acesso em: 24 abr. 2017.

REDAÇÃO OFICINA. **Segurança da informação, conceitos e mecanismos**. Disponível em: <https://www.oficinadanet.com.br/artigo/1307/seguranca_da_informacao_conceitos_e_mecanismos>. Acesso em: 25 abr. 2015.

WIKIPÉDIA. **Redundância (informática)**. Disponível em: <[https://pt.wikipedia.org/wiki/Redund%C3%A2ncia_\(inform%C3%A1tica\)](https://pt.wikipedia.org/wiki/Redund%C3%A2ncia_(inform%C3%A1tica))>. Acesso em: 25 de abril de 2017.