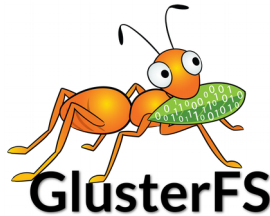


**UNIVERSIDAD LUTERANA SALVADOREÑA
FACULTAD DE CIENCIAS DEL HOMBRE Y LA NATURALEZA
LICENCIATURA EN CIENCIAS DE LA COMPUTACION
SISTEMAS OPERATIVOS DE REDES**

**SISTEMA DE FICHEROS DE ALTA DISPONIBILIDAD
CON CLUSTERFS EN UBUNTU 16.04 LTS**



PRESENTADO A:

ING MANUEL DE JESUS FLORES

PRESENTADO POR

No	APELLIDOS	NOMBRES	CARNET	PORCENTAJE PARTICIPACION
1	CLIMACO ORELLANA	DAVID ISAAC	CO01132217	100%
2	PALMA ASCENCIO	YORLENI VERENICE	PA01132489	60%

SAN SALVADOR, 02 DE JUNIO DE 2018

Tabla de Contenido

OBJETIVOS.....	1
RESUMEN (ABSTRACT).....	1
PALABRAS CLAVE.....	1
INTRODUCCION.....	2
MARCO TEORICO.....	3
Sistemas de alta disponibilidad.....	3
INCONVENIENTES.....	7
CONFIGURACION DE LOS DIFERENTES VOLUMENES EN GLUSTERFS.....	8
MATERIALES Y METODOS.....	19
Pasos generales para la instalación y configuración de GlusterFS.....	20
COMPARACION DE NUESTRO PROYECTO CON OTROS.....	20
RESULTADOS.....	23
Escenario de pruebas.....	23
CONCLUSION DE GRUPO.....	24
RECOMENDACIONES.....	24
REFERENCIA BIBLIOGRAFICA.....	25
GLOSARIO.....	28
ANEXOS.....	30
MANUAL DE INSTALACION.....	30
Configurar e instalar el Glusterfs-Cliente.....	36
GlusterFS Lado del Servidor:.....	38
GlusterFS Lado del Cliente.....	39

Indice de ilustraciones

Ilustración 1: Alta Disponibilidad.....	3
Illustration 2: Alta disponibilidad Activo - Pasivo.....	4
Ilustración 3: Alta disponibilidad Activo - Activo.....	4

Indice de Tablas

Tabla 1: Tabla comparativa y mediciones.....	22
Tabla 2: Costos del Proyecto.....	22

Indice de ilustraciones

Figura 1: Volumen Distribuido.....	8
Figura 2: Volumen replicado.....	10
Figura 3: Volumen replicado.....	11
Figura 4: Striped volume.....	13
Figura 5: Volumen Distribuido FUSE.....	15
Figura 6: User Space & Kernel.....	15
Figura 7: Geo-Replicación sobre LAN.....	16
Figura 8: Geo-replicación sobre WAN.....	17
Figura 9: Geo-replicación sobre Internet.....	17
Figura 10: Geo-replicación en cascada.....	18
Figura 11: Escenario de pruebas.....	23

OBJETIVOS

1. Implementar un cluster de servidor de ficheros de alta disponibilidad utilizando GlusterFS Server y GlusterFS cliente sobre Ubuntu 16.04 LTS
2. Configurar dos servidores de archivos que contengan la misma información y que se repliquen de manera que cuando un servidor falle, el servidor activo brinde el servicio sin interrupción del servicio.
3. Escribir los comandos necesarios vía `<<terminal>>` de Ubuntu 16.04 LTS para realizar la configuración de los servidores, cliente y conexiones de red

RESUMEN (ABSTRACT)

El proyecto aborda un sistema de Fichero de Alta Disponibilidad conocido como Gluster File System o GlusterFS, desde el punto de vista de investigación, instalación y pruebas. Para la parte de investigación, se utilizará la virtualización con Virtual Box para el entorno y los paquetes glusterfs. A través de GlusterFS, se instalará este sistema en una distribución en distintas máquinas virtuales con Sistemas operativos GNU/Linux Ubuntu Server y Ubuntu Desktop para entender cómo funciona el sistema y dar una explicación detallada de los conceptos básicos que éste envuelve como, por ejemplo, que es un volumen de almacenamiento o cuales son los mandatos básicos que se pueden utilizar. También se harán pruebas de escalabilidad, uno de los puntos destacables de Gluster FS.

Finalmente, se exponen las conclusiones obtenidas como grupo después de la realización de este proyecto, junto con el manual de instalación.

PALABRAS CLAVE

red, cliente, servidor, opensource. volumen, clúster, alta disponibilidad, GNU/Linux, glusterfs

INTRODUCCION

En éste trabajo tratamos un concepto sobre el que hemos estado trabajando durante el desarrollo del ciclo I-2018 de la Asignatura de Sistemas Operativos de redes. Servidor de Archivos de Alta disponibilidad. Cuando tenemos un sistema de almacenamiento en red, queremos que esté disponible y que los datos sean accesibles en todo momento. En el entorno empresarial, la caída de un sistema de almacenamiento puede generar pérdidas económicas sustanciales. En estos casos donde la información es vital para la continuidad del trabajo y la producción, la disponibilidad de los servicios es crítica.

“Los sistemas de alta disponibilidad garantizan los servicios de forma que si el sistema falla, sea reemplazado de forma automática sin que el cliente lo perciba”

La alta disponibilidad se basa en la **duplicación del hardware**, de esta forma, en caso de que algo falle en un sistema, tendremos otro servidor idéntico que lo sustituirá automáticamente, garantizando así la continuidad del servicio. En nuestro caso, los servidores han sido montados de manera virtual sobre VirtualBox.

Imaginemos que tenemos un Servidor en el que se almacenan los datos del programa de gestión de una empresa. Tras un fallo en su funcionamiento, es necesario sustituir un componente de dicho sistema.

Sin alta disponibilidad:

El componente puede tardar mínimo 24h en recibirse del proveedor en caso de que la tenga en stock, por lo tanto el tiempo que pasa hasta que se soluciona el problema es muy alto.

En este momento, la empresa queda parada y no puede continuar su actividad y las pérdidas económicas generadas por este parón pueden ser considerables.

Alta disponibilidad

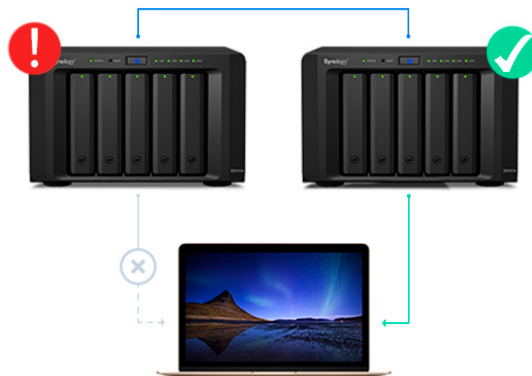


Ilustración 1: Alta Disponibilidad

MARCO TEORICO

¿Qué es GlusterFS?

GlusterFS es un sistema de archivos en red escalable y distribuido, definido para ser utilizado en el espacio de usuario, es decir, sin utilizar el espacio de almacenamiento crítico del sistema, y de esta forma no compromete el rendimiento. Se utiliza en multitud de entornos, como en análisis de datos, streaming y otras tareas intensivas de almacenamiento. Lo mejor de todo es que GlusterFS es un software gratuito y Open Source.

Un servidor de almacenamiento dispone de una máquina de las mismas características que está totalmente sincronizada y simplemente el sistema sigue funcionando con normalidad. El tiempo de respuesta de dicho sistema son minutos.

Entonces un sistema de alta disponibilidad puede ahorrar costes ante un problema de hardware.

Sistemas de alta disponibilidad

Actualmente los sistemas de alta disponibilidad cuentan con dos tecnologías:

• **Alta disponibilidad Activo – Pasivo:** Consta de dos servidores idénticos, uno de ellos activo ofreciendo los servicios y un segundo servidor, el pasivo que está en continua sincronización a la espera de una conmutación por error o programada.

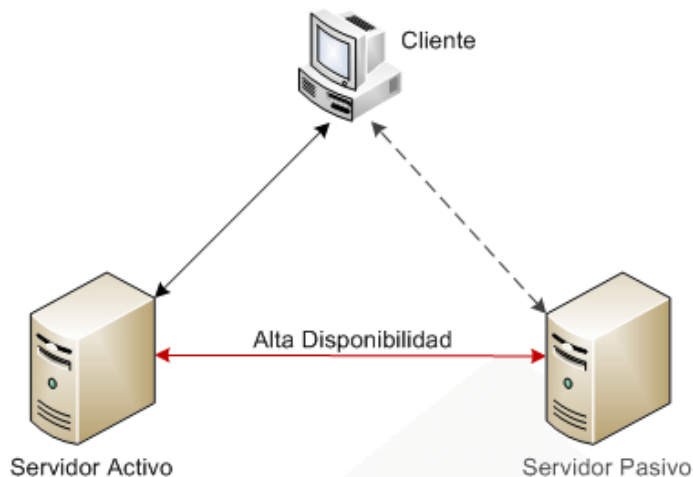


Illustration 2: Alta disponibilidad Activo - Pasivo

• **Alta disponibilidad Activo – Activo:** Consta de un servidor en el cual está todo duplicado. Este servidor es capaz de ejecutar los procesos con dos procesadores, RAM y controladoras de disco funcionando de forma simultánea y aprovechando su potencia. En caso de rotura de uno de los sistemas internos, pasaría toda la carga a uno de los procesadores.

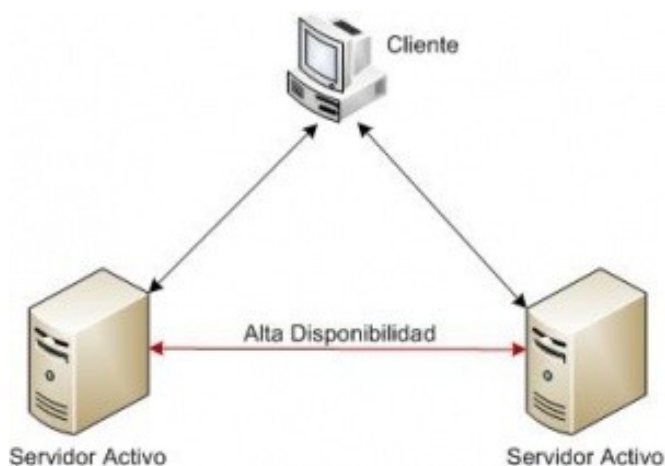


Ilustración 3: Alta disponibilidad Activo - Activo

El **Sistema de Archivos Gluster**, **Gluster File System** o **GlusterFS**, es un multiescalable sistema de archivos para NAS desarrollado inicialmente por *Gluster Inc.* Este permite agregar varios servidores de archivos sobre Ethernet o interconexiones *Infiniband*¹ *RDMA*² en un gran entorno de archivos de red en paralelo.

El diseño del GlusterFS se basa en la utilización del espacio de usuario y de esta manera no compromete el rendimiento. Se pueden encontrar siendo utilizado en una gran variedad de entornos y aplicaciones como computación en nube, ciencias biomédicas y almacenamiento de archivos, Streaming, Servidores web, etc. GlusterFS está licenciado bajo **GNU General Public License versión 3**.

Gluster Inc. fue el principal patrocinador comercial del GlusterFS, el cual ofrece tanto productos comerciales como apoyo para desarrollo de soluciones libres basadas en el GlusterFS.

GlusterFS se basa en la interacción de componentes cliente y servidor. Los servidores normalmente se implementan como almacenamiento en bloques, en cada servidor el proceso **daemon glusterfsd** exporta un sistema de archivos local como un volumen.

El proceso cliente Glusterfs, se conecta a los servidores a través de algún protocolo **TCP/IP**, **InfiniBand** o **SDP**, compone volúmenes compuestos virtuales a partir de los múltiples servidores remotos, mediante el uso de traductores.

Por defecto, los archivos son almacenados enteros, pero también puede configurarse que se fragmente en múltiples porciones en cada servidor. Los volúmenes pueden ser montados en los equipos cliente mediante el modulo FUSE o acceder a través de la librería cliente *libglusterfs* sin incurrir en problemas con el sistema de archivos FUSE.

1 **InfiniBand** es un bus de comunicaciones serie de alta velocidad, baja latencia y de baja sobrecarga de CPU, diseñado tanto para conexiones internas como externas

2 En computación, remote direct memory access (RDMA) o su traducción acceso remoto directo a memoria consiste en el acceso directo desde la memoria principal de un ordenador en la de otro sin cooperación del sistema operativo

La mayor parte de la funcionalidades del GlusterFS se implementa como traductores, incluyendo:

- Espejado y la replicación de archivos.
- Fragmentación de los archivos o Data striping.
- Balanceo de carga para la lectura y escritura de archivos.
- Volúmenes con tolerancia a fallos.
- Planificación de E/S y almacenamiento en caché de disco.
- Las cuotas de almacenamiento

El servidor GlusterFS se mantiene mínimamente simple: exporta un sistema de archivos existente como está, dejando en manos de los traductores del lado del cliente a la estructura del almacenamiento. Los propios clientes se manejan independientemente, no se comunican directamente entre sí, y los traductores administran la consistencia de los datos entre ellos.

GlusterFS se basa en un algoritmo de *hash elástico* en vez de utilizar un modelo de metadatos centralizados o distribuidos. Desde la versión 3.1, los volúmenes pueden ser agregados, eliminados o migrados en forma dinámica, esto ayuda a prever problemas de consistencia, y permite que el GlusterFS pueda ser escalado a varios petabytes sobre hardware de bajo coste, evitando así los cuellos de botella que normalmente afectan a muchos sistemas de archivos distribuidos con múltiple concurrencia.

Ventajas de GlusterFS

- **Simplicidad:** Es fácil de utilizar y al ser ejecutado en el espacio de usuario es independiente del núcleo.
- **Elasticidad:** Se adapta al crecimiento y reduce el tamaño de los datos.
- **Escalabilidad:** Tiene disponibilidad de Petabytes y más.
- **Velocidad:** Elimina los metadatos y mejora el rendimiento considerablemente unificando los datos y objetos.
- **Gratuito**
- **Licencia GNU/GPL**
- **Fácil de configurar**
- **Comunicación por RED y por SERIE Sistemas heterogéneos**

- **Altamente soportado**
- **Integrable con otras aplicaciones**

Sistemas en los que se puede instalar

- CentOS
- Debian
- Fedora
- RHEL
- SuSE
- Ubuntu

OBJETIVOS DE GLUSTERFS

1. Asegurar actividad
2. Redundancia en comunicaciones de control Detectar caídas del sistema
3. Recuperar actividades muertas
4. Takeover (Intercambiar IP)

CARACTERISTICAS DE GLUSTERFS

- Comunicar su estado a los otros nodos Comprobar estado de los nodos
- Capacidad de suplantar las actividad de los nodos caídos
- Capacidad de ocupar la IP asignada al servicio
- Tomar acciones extras ante caídas
- Tiempo de recuperación de 30 a 60 segundos

INCONVENIENTES

- Nodo de reserva desaprovechado
- Pérdida de conexiones remotas
- Obliga a centralizar el sistema de ficheros

CONFIGURACION DE LOS DIFERENTES VOLUMENES EN GLUSTERFS

El volumen es la colección de bloques (bricks) y la mayoría de las operaciones del sistema de archivos Gluster se realizan en el volumen.

El sistema de archivos Gluster admite diferentes tipos de volúmenes según los requisitos. Algunos volúmenes son buenos para escalar el tamaño de almacenamiento, algunos para mejorar el rendimiento y otros para ambos.

1. Volumen distribuido: este es el volumen glusterfs predeterminado, es decir, al crear un volumen, si no especifica el tipo de volumen, la opción predeterminada es crear un volumen distribuido.

Aquí, los archivos se distribuyen a través de varios bloques (bricks) en el volumen. Por lo tanto, el archivo 1 se puede almacenar solo en brick1 o brick2, pero no en ambos. Por lo tanto, no hay redundancia de datos. El propósito de dicho volumen de almacenamiento es escalar fácilmente y de manera económica el tamaño del volumen. Sin embargo, esto también significa que una falla de bloques dará lugar a la pérdida completa de datos y uno debe confiar en el hardware subyacente para la protección contra la pérdida de datos.

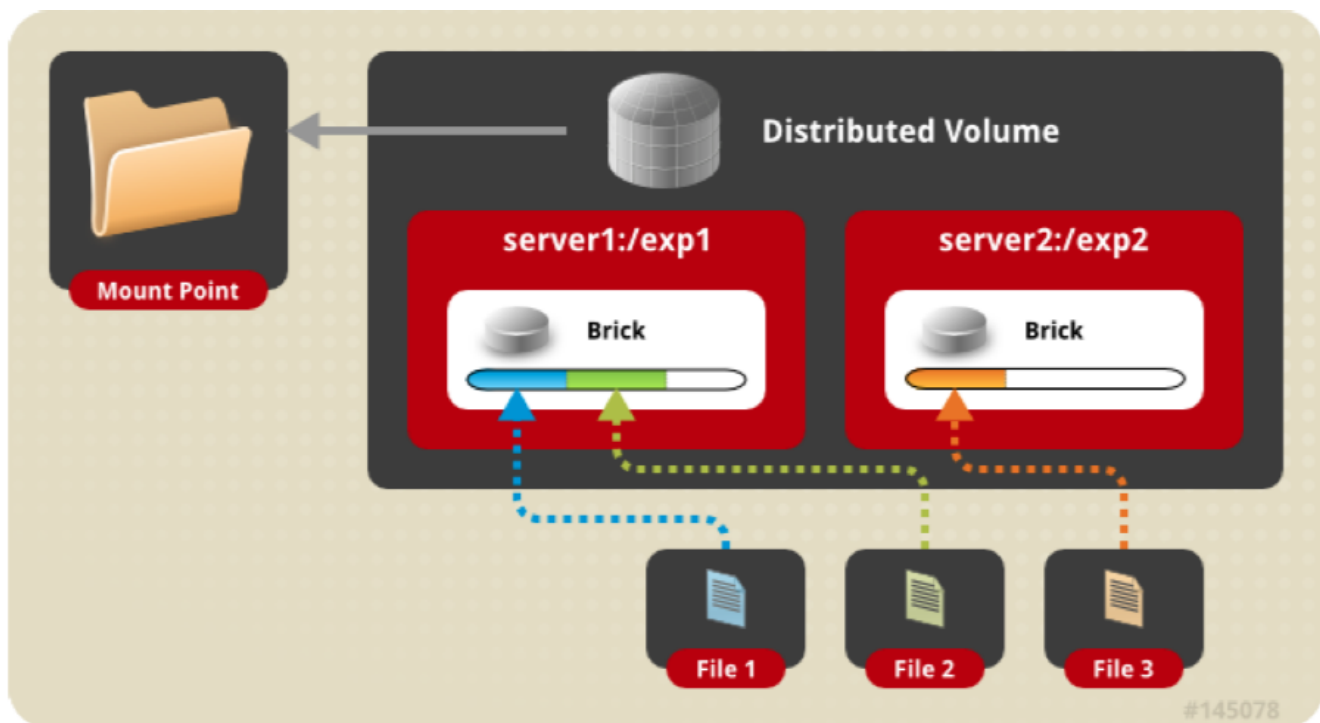


Figura 1: Volumen Distribuido

Crear un volumen distribuido

gluster volume create NEW-VOLNAME [transport [tcp | rdma | tcp,rdma]] NEW-BRICK...

Por ejemplo, para crear un volumen distribuido con cuatro servidores de almacenamiento usando TCP.

```
# gluster volume create test-volume server1:/exp1 server2:/exp2 server3:/exp3 server4:/exp4
↪ Creation of test-volume has been successful
Please start the volume to access data
```

Para mostrar la información del volumen

```
# gluster volume info
Volume Name: test-volume
Type: Distribute
Status: Created
Number of Bricks: 4
Transport-type: tcp
Bricks:
Brick1: server1:/exp1
Brick2: server2:/exp2
Brick3: server3:/exp3
Brick4: server4:/exp4
```

2. Volumen replicado: en este volumen superamos el problema de pérdida de datos que enfrenta el volumen distribuido. Aquí se guardan copias exactas de los datos en todos los bloques (bricks). El cliente puede decidir la cantidad de réplicas en el volumen mientras crea el volumen. Entonces, necesitamos tener al menos dos lBricks para crear un volumen con 2 réplicas o un mínimo de tres Bricks para crear un volumen de 3 réplicas. Una ventaja importante de tal volumen es que incluso si un brick falla, se puede acceder a los datos desde sus bricks replicados. Tal volumen se usa para una mejor confiabilidad y redundancia de datos.

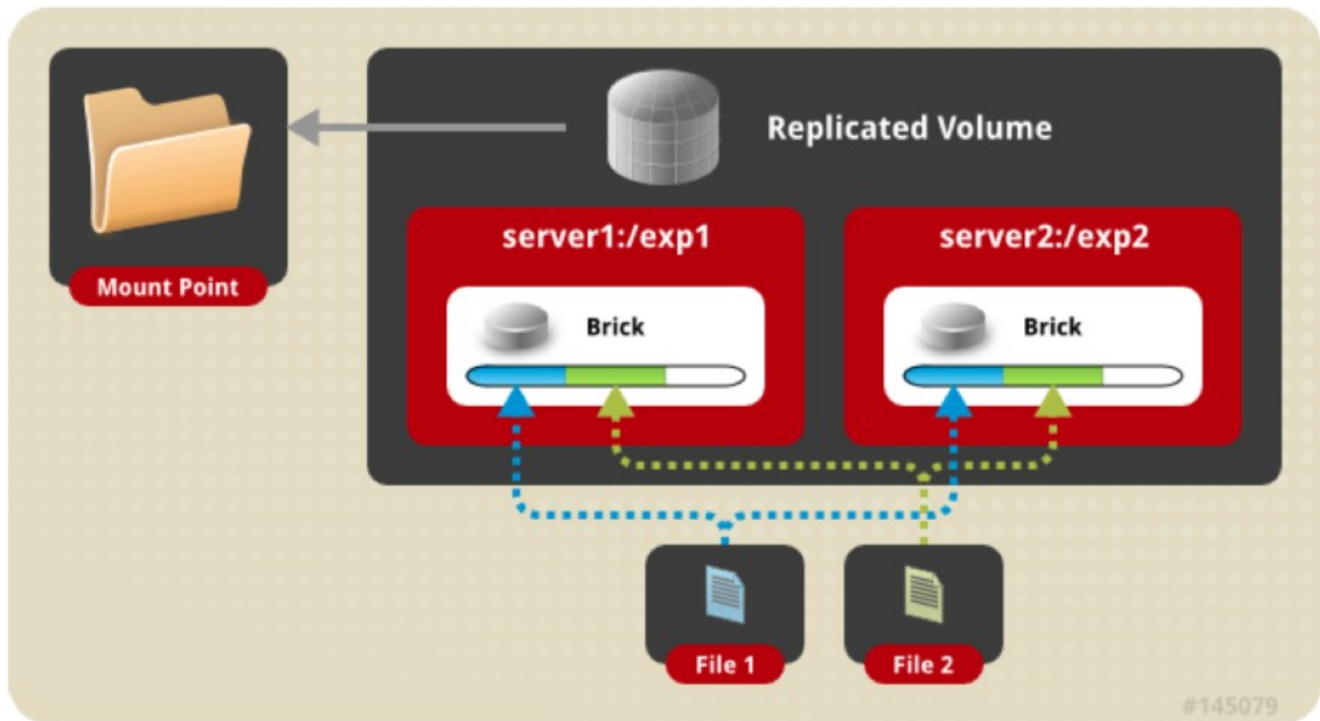


Figura 2: Volumen replicado

Crear un volumen replicado

`gluster volume create NEW-VOLNAME [replica COUNT] [transport [tcp | rdma | tcp,rdma]] NEW-BRICK...`

Por ejemplo, para crear un volumen replicado con dos servidores de almacenamiento:

```
# gluster volume create test-volume replica 2 transport tcp server1:/exp1 server2:/  
exp2  
Creation of test-volume has been successful  
Please start the volume to access data
```

3.0 Volumen replicado distribuido: en este volumen, los archivos se distribuyen entre conjuntos de bricks replicados. La cantidad de bricks debe ser un múltiplo del recuento de réplicas. También importa el orden en que especificamos los bricks, ya que los bricks adyacentes se convierten en réplicas entre sí. Este tipo de volumen se usa cuando se requiere una alta disponibilidad de datos debido a la redundancia y el almacenamiento a escala. Entonces, si hubo ocho bricks y la réplica cuenta 2, los primeros dos bricks se convierten en réplicas del otro, luego los siguientes dos y así sucesivamente. Este volumen se denota como 2x2. De manera similar, si hubo ocho bricks y la réplica cuenta 4, entonces cuatro bricks se convierten en réplica uno del otro y denotamos este volumen como volumen de 2x4.

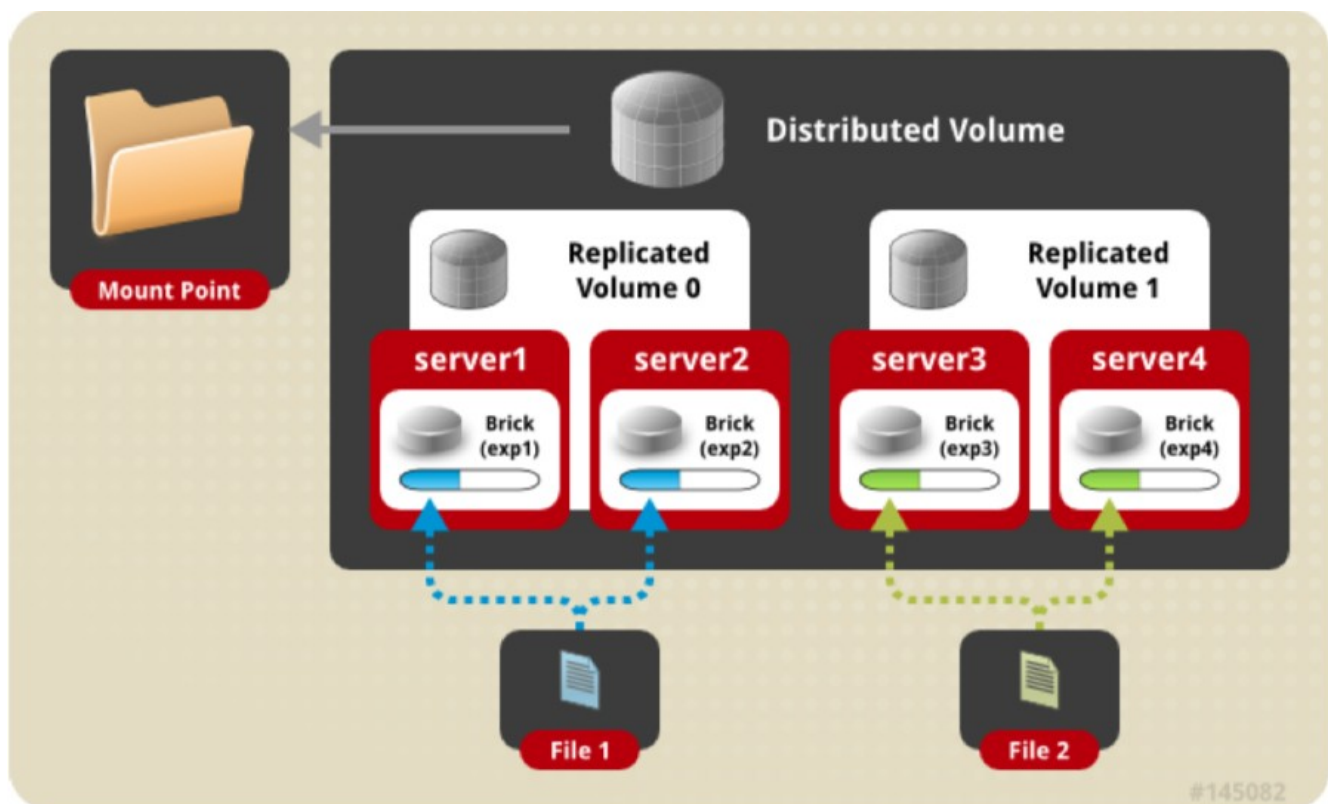


Figura 3: Volumen replicado

Crear el volumen replicado distribuido

gluster volume create NEW-VOLNAME [replica COUNT] [transport [tcp | rdma | tcp,rdma]] NEW-BRICK...

Por ejemplo, volumen distribuido (replicado) de cuatro nodos con un espejo bidireccional:

```
# gluster volume create test-volume replica 2 transport tcp server1:/exp1 server2:/  
exp2 server3:/exp3 server4:/exp4  
Creation of test-volume has been successful  
Please start the volume to access data
```

4.0 Volumen rayado (Striped Volume): considere la posibilidad de almacenar un archivo grande en un bloque al que accedan frecuentemente muchos clientes al mismo tiempo. Esto causará demasiada carga en un solo brick y reduciría el rendimiento. En el volumen seccionado, los datos se almacenan en los bricks después de dividirlos en diferentes franjas. Por lo tanto, el archivo grande se dividirá en fragmentos más pequeños (igual al número de bloques en el volumen) y cada fragmento se almacenará en un bloque. Ahora la carga se distribuye y el archivo se puede recuperar más rápido pero no se proporciona redundancia de datos.

Crear un Striped Volume:

#gluster volume create NEW-VOLNAME [stripe COUNT] [transport [tcp | dma | tcp,rdma]] NEW-BRICK...

Por ejemplo, para crear un volumen seccionado en dos servidores de almacenamiento:

```
# gluster volume create test-volume stripe 2 transport tcp server1:/exp1 server2:/exp2  
Creation of test-volume has been successful  
Please start the volume to access data
```

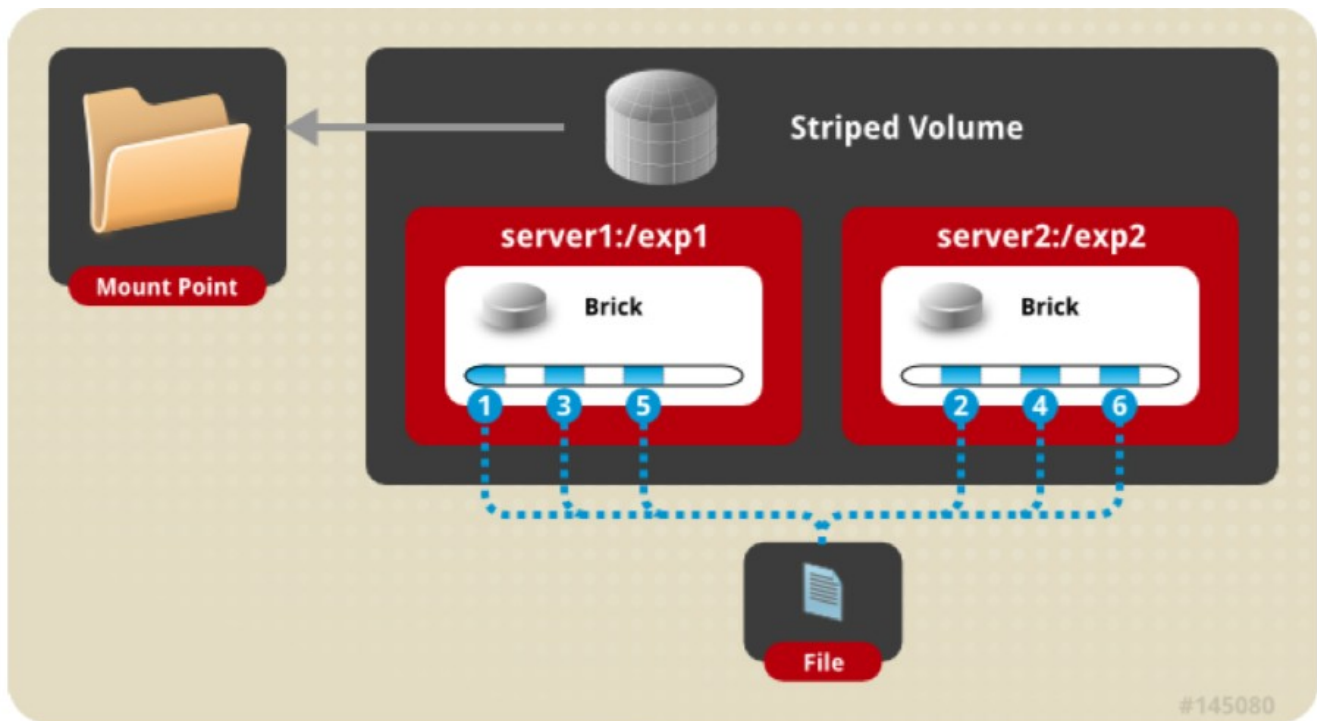


Figura 4: Striped volume

5.0 Volumen de rayas distribuido (Distributed Striped Volume): es similar al volumen de Striff Glusterfs, excepto que ahora las rayas se pueden distribuir a través de más bricks. Sin embargo, el número de bricks debe ser un múltiplo del número de rayas. Entonces, si queremos aumentar el tamaño del volumen, debemos agregar ladrillos en el múltiplo del recuento de bandas.

Crea el volumen distribuido rayado:

```
gluster volume create NEW-VOLNAME [stripe COUNT] [transport [tcp | rdma | tcp,rdma]] NEW-BRICK...
```

Por ejemplo, para crear un volumen distribuido seccionado en ocho servidores de almacenamiento:

```
# gluster volume create test-volume stripe 4 transport tcp server1:/exp1 server2:/
→exp2 server3:/exp3 server4:/exp4 server5:/exp5 server6:/exp6 server7:/exp7 server8:/
→exp8
Creation of test-volume has been successful
Please start the volume to access data.
```


6.0 FUSE

GlusterFS es un sistema de archivos de espacio de usuario. Esta fue una decisión tomada por los desarrolladores de GlusterFS inicialmente, ya que obtener los módulos en Linux Kernel es un proceso muy largo y difícil.

Siendo un sistema de archivos de espacio de usuario, para interactuar con VFS del kernel, GlusterFS hace uso de FUSE (File System in Userspace) cuya traducción es: Sistema de archivos en el espacio de usuario. Durante mucho tiempo, la implementación de un sistema de archivos de espacio de usuario se consideró imposible. FUSE se desarrolló como una solución para esto. FUSE es un módulo kernel que admite la interacción entre VFS del kernel y aplicaciones de usuario no privilegiadas, y tiene una API a la que se puede acceder desde el espacio de usuario. Al usar esta API, se puede escribir cualquier tipo de sistema de archivos usando casi cualquier idioma que prefiera, ya que hay muchas vinculaciones entre FUSE y otros idiomas.

Diagrama estructural de FUSE

Esto muestra un sistema de archivos "hello world" que se compila para crear un "hola" binario. Se ejecuta con un punto de montaje del sistema de archivos / tmp / fuse. Luego, el usuario emite un comando ls -l en el punto de montaje / tmp / fuse. Este comando llega a VFS a través de glibc y dado que mount / tmp / fuse corresponde a un sistema de archivos basado en FUSE, VFS lo pasa al módulo FUSE. El módulo de kernel de FUSE se pone en contacto con el sistema de archivos actual "hola" binario después de pasar por glibc y la biblioteca FUSE en el espacio de usuario (libfuse). El resultado es devuelto por el "saludo" a través de la misma ruta y alcanza el comando ls -l.

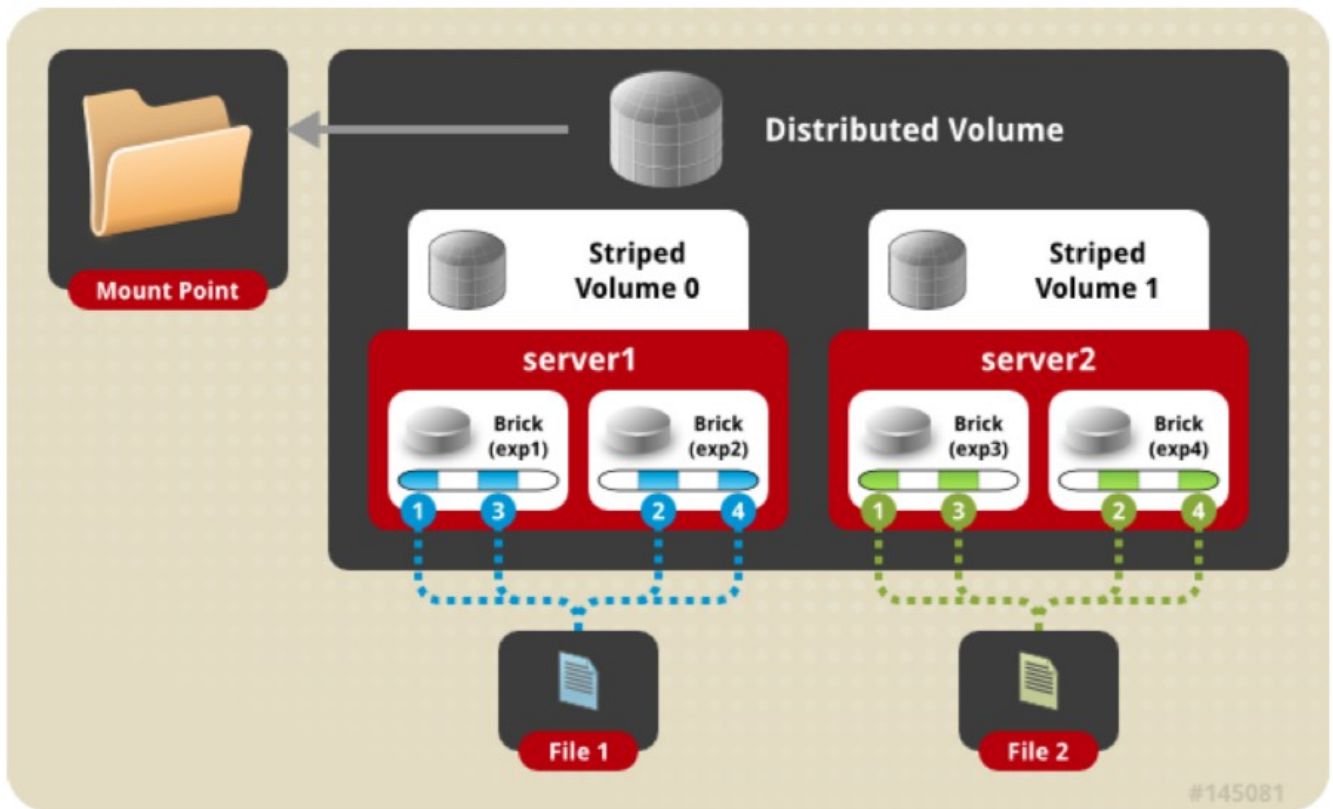


Figura 5: Volumen Distribuido FUSE

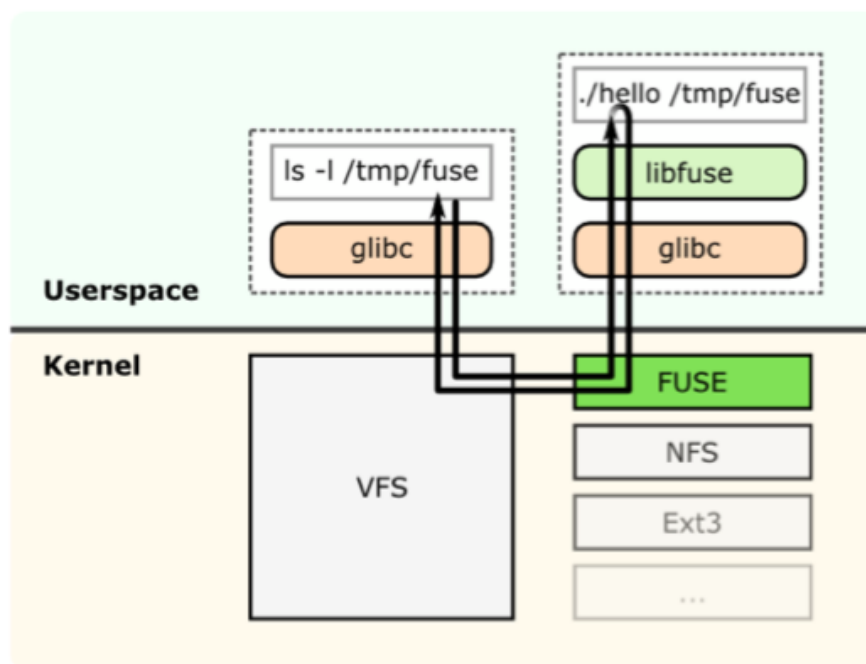


Figura 6: User Space & Kernel

La comunicación entre el módulo de kernel de FUSE y la biblioteca de FUSE (libfuse) se realiza a través de un descriptor de archivo especial que se obtiene al abrir / dev / fuse. Este archivo se puede abrir varias veces y el descriptor de archivo obtenido se pasa a mount syscall para hacer coincidir el descriptor con el sistema de archivos montado.

7.0 Geo-Replicación

La Geo-replicación proporciona replicación asíncrona de datos en ubicaciones geográficamente distintas y se introdujo en Glusterfs 3.2. Principalmente funciona en WAN y se utiliza para replicar todo el volumen a diferencia de AFR, que es la replicación dentro del clúster. Esto es principalmente útil para la copia de seguridad de datos completos para recuperación de desastres.

La Geo-replicación usa un modelo maestro-esclavo, por el cual se produce la replicación entre el Máster, un volumen GlusterFS y el Esclavo, que puede ser un directorio local o un volumen GlusterFS. El esclavo (se accede al directorio local o al volumen usando el túnel SSH).

La geo replicación proporciona un servicio de replicación incremental a través de redes de área local (LAN), red de área amplia (WAN) y en Internet.

Geo-replicación en LAN

Puede configurar Geo-replicación para reflejar datos a través de una red de área local.

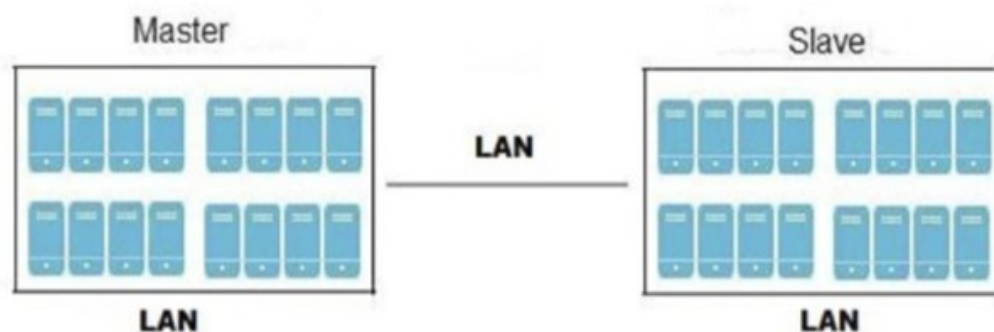


Figura 7: Geo-Replicación sobre LAN

Geo-replicación sobre WAN

Puede configurar Geo-replicación para replicar datos a través de una red de área amplia



Figura 8: Geo-replicaicón sobre WAN

Geo-replicación a través de Internet

Puede configurar la Geo-replicación para reflejar datos a través de Internet.



Figura 9: Geo-replicación sobre Internet

Geo-replicación en cascada de varios sitios

Puede configurar la Geo-replicación para duplicar datos en forma de cascada en múltiples sitios.

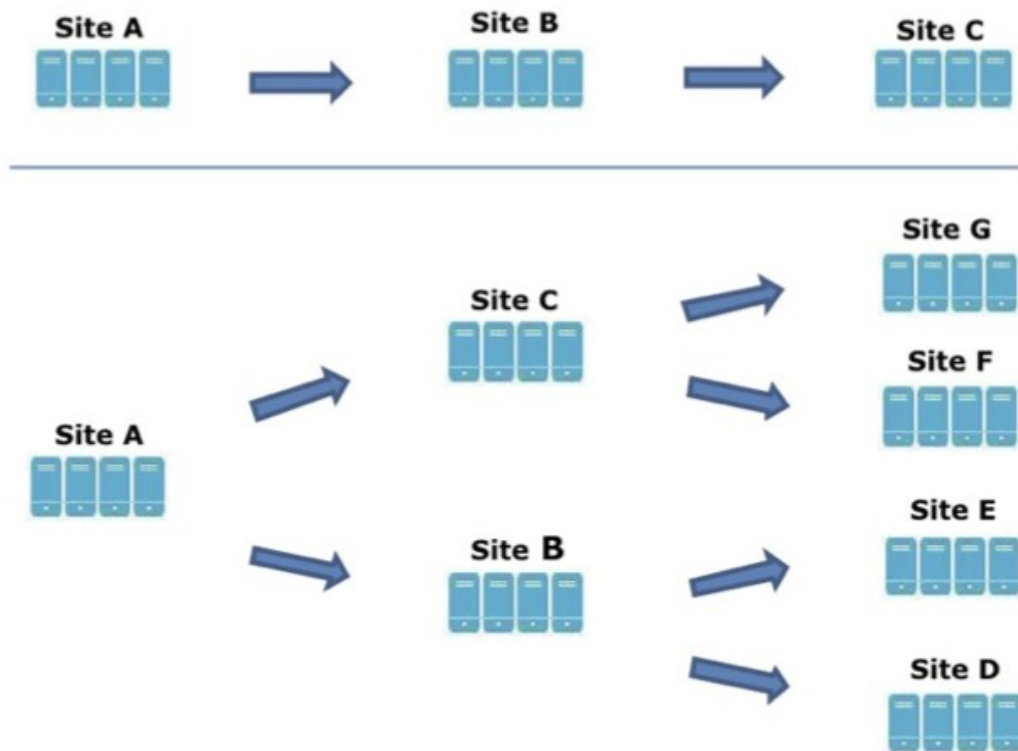


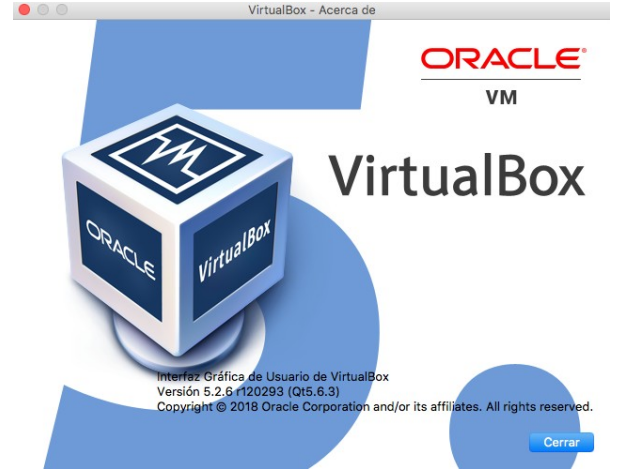
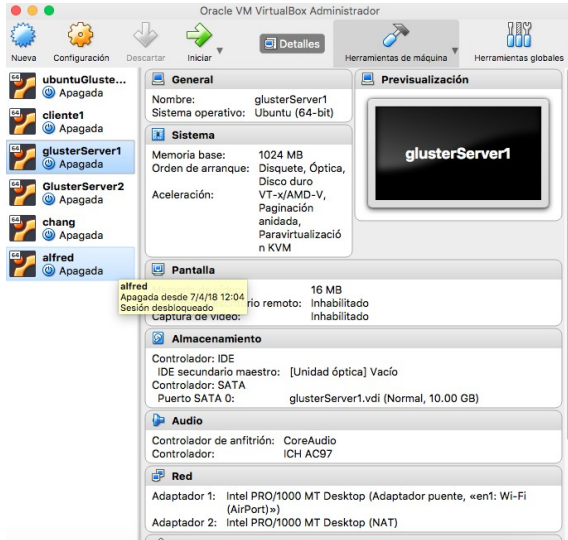
Figura 10: Geo-replicación en cascada

MATERIALES Y METODOS

Para la elaboración de nuestro proyecto se fue necesario utilizar:

1. Tres computadoras (o maquinas virtuales), un cliente y dos servidores

Montados sobre Virtual Box versión 5.2.8 de Oracle



2. Sistema Operativo GNU/Linux Ubuntu 16.04 LTS

con entorno gráfico montadas sobre Virtual Box con 10 Gb de disco duro, memoria de 1024 MB

3. Switch que interconecte los equipos. Nexxt para la iteracción de los equipos y conexión con el exterior

4. Un cable de red RJ45 para conexión con el switch

Pasos generales para la instalación y configuración de GlusterFS

(Se pueden encontrar el paso a paso a detalle en el manual de instalación de GlusterFS sobre Ubuntu 16.04 LTS)

1. Instalar las maquinas virtuales sobre VirtualBox, a partir de una imagen ISO Ubuntu 16.04 LTS

2. Arrancar la máquina virtual1, nombrada como GlusterServer1 Y GlusterServer2

2.1 Abrir terminal

2.2 Instalar GlusterFS Server

2.3 Modificar el archivo de configuración de glusterFS Server

2.4 Establecer las conexiones y verificar comunicacion

3. Arrancar la maquina virtual nombrada GlusterCliente

3.1 Abrir terminal

3.2 Instalar GlusterFS Client y verificar comunicaciones

4. Realizar pruebas de conexión con metodo Activo- Activo, Activo - Pasivo

ASPECTOS QUE SE TOMARON EN CUENTA EN LA COMPARACION DE NUESTRO PROYECTO CON OTROS.

- El Proyecto consistia en Instalar, configurar y probar el funcionamiento de GlusterFS, sobre Ubuntu 16.04 LTS, por ello debíamos conocer que otras alternativas de software existian, que brindaran Alta Disponibilidad en Archivos, y nos encontramos con muchos otros Software tanto opensource como privativo que realizan la misma tarea, verificamos sus características principales, la facilidad de configuración, el bajo costo, la cantidad de pasos para su instalación y constatamos que GlusterFS ofrece mucha documentación en linea para solventar cualquier aspecto de configuracion e instalación, esto facilitó el desarrollo del proyecto y aseguró su funcionamiento.

El cuadro 1, resume los aspectos que se compararon durante le ejecución de la etapa de investigación.

<i>SISTEMA</i>	<i>Heartbeats</i>	<i>GlusterFS</i>	<i>CODA</i>	<i>DRBD</i>
<i>Características</i>	<p><i>Necesita emplearse en combinación con un gestor de recursos del clúster (cluster resource manager – CRM)</i></p> <p><i>Puede configurarse para trabajar de forma pasiva y/o activa</i></p>	<p><i>Capacidad de suplantar las actividad de los nodos caídos</i></p> <p><i>Capacidad de ocupar la IP asignada al servicio</i></p> <p><i>Tomar acciones extras ante caídas</i></p> <p><i>Tiempo de recuperación de 30 a 60 segundos</i></p>	<p><i>Puede funcionar sin conexión</i></p> <p><i>Es software libre</i></p> <p><i>Gran rendimiento gracias a la caché persistente en el cliente</i></p> <p><i>Replicado de servidores</i></p> <p><i>Modelo de seguridad para autenticación, cifrado y control de acceso</i></p> <p><i>Funcionamiento continuado durante fallos de red</i></p>	<p><i>Replicación Distribuida de Dispositivos de Bloques</i></p> <p><i>Copia de seguridad en tiempo real</i></p> <p><i>Sistema de transacciones a nivel del sistema de ficheros</i></p> <p><i>Automatismo en la sincronización</i></p>
<i>Ventajas</i>	<p><i>Fácil de configurar</i></p> <p><i>Comunicación</i></p> <p><i>Puede trabajar con tantos nodos como se requiera.</i></p>	<p><i>Fácil de configurar</i></p> <p><i>Comunicación por RED y por SERIE</i></p> <p><i>Sistemas heterogéneos</i></p> <p><i>Altamente soportado</i></p> <p><i>Integrable con otras aplicaciones</i></p>	<p><i>Realiza la operación de escritura en segundo plan</i></p> <p><i>Soporta ACLs (Access Control List) Acceso con lista de control</i></p>	<p><i>Gratuito y con licencia GNU/GPL</i></p> <p><i>Replicación de datos automática</i></p> <p><i>Funcionamiento remoto</i></p> <p><i>Fácil de configurar</i></p> <p><i>Abstracción con el sistema operativo</i></p> <p><i>Acceso remoto fácil.</i></p>

<i>SISTEMA</i>	<i>Heartbeats</i>	<i>GlusterFS</i>	<i>CODA</i>	<i>DRBD</i>
<i>Desventajas</i>	<i>Se debe desactivar desactivar el cortafuegos predeterminando del sistema en ambos nodos, debido a que éste interfiere con la comunicación entre los nodos de heartbeat</i>	<i>Nodo de reserva desaprovechado</i> <i>Pérdida de conexiones remotas</i> <i>Obliga a centralizar el sistema de ficheros</i>	<i>Las operaciones de bloqueo de ficheros no están implementadas</i> <i>Los volúmenes Coda no son redimensionables.</i>	
<i>Costo</i>	<i>Gratuito</i>	<i>Gratuito</i>	<i>Gratuito</i>	<i>Gratuito</i>

Tabla 1: Tabla comparativa y mediciones

COSTOS DEL PROYECTO

<i>Cant</i>	<i>DESCRIPCION</i>	<i>COSTO \$</i>
<i>1</i>	<i>Computadora MAC, i5 4Gb de RAM.</i>	<i>\$ 1,200.00</i>
<i>1</i>	<i>Switch Nexxt Wireless, y conexión cableada</i>	<i>\$ 25.00</i>
<i>1</i>	<i>Cable RJ45</i>	<i>\$ 6.00</i>
<i>2</i>	<i>Horas técnicos, investigación y configuracion (48 horas c/u)</i>	<i>\$ 600.00</i>
<i>2</i>	<i>Viáticos, alimentación, transporte, internet.</i>	<i>\$ 100.00</i>
<i>1</i>	<i>Papeleria, equipo, anillados, impresiones</i>	<i>\$ 45.00</i>
<i>1</i>	<i>Otros gastos 10% extra de la suma total</i>	<i>\$ 197.60</i>
	<i>TOTAL</i>	<i>\$ 2,083.60</i>

Tabla 2: Costos del Proyecto

RESULTADOS

Escenario de pruebas

Diagrama de conexiones

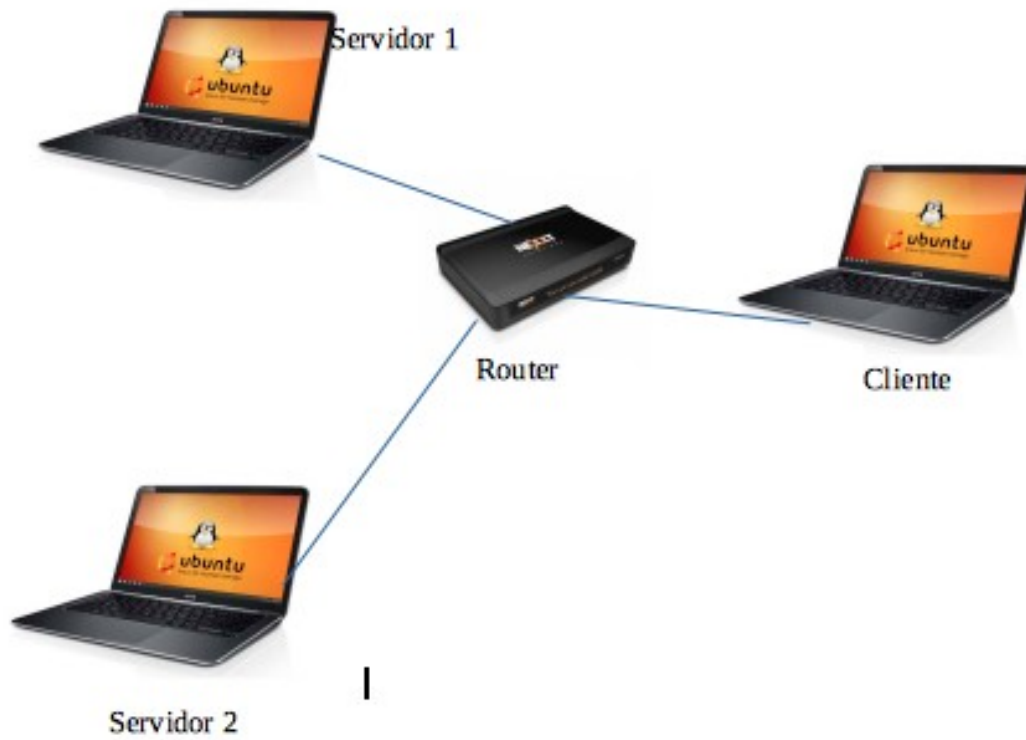


Figura 11: Escenario de pruebas

CONCLUSION DE GRUPO

Como era de esperarse nuestro proyecto se ha configurado muy bien y es funcional, ya que al realizar las pruebas y desactivar el Gluster1, el Gluster 2 conserva y sirve la información requerida por el cliente y viceversa.

Lo que no pudimos verificar fue la alta disponibilidad, suponemos que esto requeriria que mas clientes se conecten al o los servidores para asegurarnos que ante la caída de el server 1 o el server 2, siempre el cliente obtenga lo que solicita. La falta de recursos físicos en los equipos tales como memoria RAM y Procesador mas potente no nos permite realizar estas pruebas.

Durante el proceso de comparaciones, se instaló y configuró Heartbeat, con ello se realizaron pruebas de escalabilidad, rendimiento y características, el cual es de fácil instalación y podemos recomendar utilizar heartbeat como alternativa para un Sistema de Ficheros Distribuidos de alta disponibilidad.

GlusterFS, requiere menos pasos e instrucciones en su configuración, por eso consideramos que es práctico y fácil manejo de parámetros para su funcionamiento.

RECOMENDACIONES

1. Seguir el manual de instalación paso a paso para obtener buenos resultados
2. Montarlo sobre Ubuntu 16.04 LTS o Ubuntu Server
3. Familiarizarse con las instrucciones en terminal de los sistemas GNU/Linux
4. Familiarizarse con el uso de terminal para la configuración
5. Poseer previa experiencia en la instalación de Sistemas Operativos en VirtualBox
6. Poseer cierta habilidad y conocimiento de configuración de IP Staticas en sistemas GNU/Linux
7. Poseer deseos y mucha paciencia en la configuración de GlusterFS

Existe una gran variedad dentro de GNU/Linux de Sistemas de Archivos Distribuidos de Alta Disponibilidad, útiles herramientas de fácil instalación y configuración para los usuarios y administradores y además de Libre Distribución y sin costo.

REFERENCIA BIBLIOGRAFICA

URL	TITULO	FECHA CONSULTA	AUTOR
http://www.monografias.com/trabajos74/sistema-archivos-distribuidos-coda/sistema-archivos-distribuidos-coda2.shtml#ixzz5CNSZNTxR	Sistema de archivos distribuidos: CODA (Sistemas Operativos)	19 – 02 -2018	Leonardo Sarango
http://redesteleco.com/drbd-raid1_en_red_entre_varios_equipos/	DRBD: RAID1 en red entre varios equipos	20 – 02 -2018 30 – 05 - 2018	--
https://mviera.io/blog/compartiendo-ficheros-con-glusterfs/	Compartiendo ficheros con GlusterFS	19 – 02 -2018 20 – 02 -2018	MANUEL VIERA
https://es.wikipedia.org/wiki/ReiserFS#Desventajas	ReiserFS	21 – 02- 2018	Wikipedia®
https://www.proxadmin.es/blog/glusterfs-almacenamiento-distribuido/	GLUSTERFS: CREATU ALMACENAMIENTO DISTRIBUIDO	24 -02-2018	Proxadmin
https://qloudea.com/blog/alta-disponibilidad-servidor-almacenamiento/	Alta disponibilidad en Servidor de almacenamiento	15 – 03 -2018	David Aragón.
https://jonathan.vargas.cr/es/temas/plataforma/18291-almacenamiento-de-alta-disponibilidad-con-glusterfs	<u>Almacenamiento en Alta Disponibilidad con GlusterFS</u>	15 – 03 -2018 28 – 03 -2018	Jonathan Vargas

http://plataformas.uls.edu.sv/pluginfile.php/10033/mod_resource/content/4/sor-2018-proyectos.pdf	Asignación de proyecto Final	Febrero - Mayo	Ing. Manuel Flores
https://www.howtoforge.com/high-availability-storage-with-glusterfs-3.2.x-on-debian-wheezy-automatic-file-replication-mirror-across-two-storage-servers	High-Availability Storage With GlusterFS 3.2.x On Debian Wheezy - Automatic File Replication (Mirror) Across Two Storage Servers	02 – 04 -2018 30 – 05 - 2018	
https://sigterm.sh/2014/02/01/highly-available-nfs-cluster-on-debian-wheezy/	Highly Available NFS Cluster on Debian Wheezy	02– 04 -2018 06– 04 -2018	Gregory Armer
https://lsub.org/ls/export/pfc_pfs/node11.html	Sistemas de ficheros distribuidos	23– 04 -2018	<i>Jesus Vergara Igual</i>
https://media.readthedocs.org/pdf/glusterdocs-beta/latest/glusterdocs-beta.pdf	GlusterFS Documentation Release 3.8.0	23– 04 -2018 30 – 05 - 2018	Gluster Community
https://www.gluster.org	Download Glusterfs	02– 04 -2018 06– 04 -2018	Gluster org
https://docs.gluster.org/en/latest/Install-Guide/Overview/	Install Guide	10– 04 -2018	
https://github.com/gluster/glusterdocs/	Gluster Docs	10– 04 -2018 30 – 05 - 2018	Github Community

https://www.youtube.com/watch?v=UxyPLnlCdhA&t=10s	Gluster Setup...The Easy Way	10- 04 -2018	Paul Cuzner
https://www.youtube.com/watch?v=NYGn7sgMrMw	Gluster Storage Replication	02- 04 -2018 10- 04 -2018 12- 04 -2018	Linux King
https://www.youtube.com/watch?v=KZwKFY3niEs	GlusterFS – How to Create a Distributer GlusterFS Volumes	22 -02-2018 30 – 05 - 2018	Yalla Labs
https://www.youtube.com/watch?v=I7Utpv4ORE4	GlusterFS Pada ubuntu Server 16.04	19 – 02 -2018 20 – 02 -2018 21 – 02 -2018	Anshori Muhammad
https://www.youtube.com/watch?v=iyPnTHh-9E4	Gluster Hadoop	22 -02- 2018 30 – 05 - 2018	Arturo martin romero

GLOSARIO

Alta Disponibilidad (High availability) :Es un protocolo de diseño del sistema y su implementación asociada que asegura un cierto grado absoluto de continuidad operacional durante un período de medición dado. **Disponibilidad** se refiere a la habilidad de la comunidad de usuarios para acceder al sistema, someter nuevos trabajos, actualizar o alterar trabajos existentes o recoger los resultados de trabajos previos.

Brick: Es el almacenamiento básico (directorio) en un servidor en el grupo de almacenamiento de confianza.

Clúster: El término clúster (*del inglés cluster, que significa grupo o racimo*) se aplica a los conjuntos o conglomerados de ordenadores unidos entre sí normalmente por una red de alta velocidad y que se comportan como si fuesen una única computadora.

Cliente: Un cliente realiza peticiones a otro programa, el servidor, quien le da respuesta. Esta idea también se puede aplicar a programas que se ejecutan sobre una sola computadora, aunque es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras.

GNU/Linux: GNU/Linux, también conocido como Linux, es un sistema operativo libre tipo Unix; multiplataforma, multiusuario y multitarea. El sistema es la combinación de varios proyectos, entre los cuales destacan GNU (encabezado por Richard Stallman y la Free Software Foundation) y el núcleo Linux (encabezado por Linus Torvalds). Su desarrollo es uno de los ejemplos más prominentes de software libre: todo su código fuente puede ser utilizado, modificado y redistribuido libremente por cualquiera, bajo los términos de la GPL (Licencia Pública General de GNU) y otra serie de licencias libres.

Opensource: El código abierto es un modelo de desarrollo de software basado en la colaboración abierta. Se enfoca más en los beneficios prácticos (acceso al código fuente) que en cuestiones éticas o

de libertad que tanto se destacan en el software libre. Para muchos el término «libre» hace referencia al hecho de adquirir un software de manera gratuita, pero más que eso, la libertad se refiere al poder modificar la fuente del programa sin restricciones de licencia, ya que muchas empresas de software encierran su código, ocultándolo, y restringiéndose los derechos a sí misma.

Red: Una red de computadoras (también llamada red de ordenadores, red de comunicaciones de datos, red informática) es un conjunto de equipos informáticos y software conectados entre sí por medio de dispositivos físicos o inalámbricos que envían y reciben impulsos eléctricos, ondas electromagnéticas o cualquier otro medio para el transporte de datos, con la finalidad de compartir información, recursos y ofrecer servicios

Replicar: Hacer copias múltiples de datos para lograr una alta redundancia.

Servidor: Un servidor es una aplicación en ejecución (software) capaz de atender las peticiones de un cliente y devolverle una respuesta en concordancia. Los servidores se pueden ejecutar en cualquier tipo de computadora, incluso en computadoras dedicadas a las cuales se les conoce individualmente como «el servidor».

Volumen: En informática, un volumen de almacenamiento es una división lógica en un medio de almacenamiento, para nuestro caso el Disco Duro Virtual

VirtualBox: Oracle VM VirtualBox es un software de virtualización para arquitecturas x86/amd64. Actualmente es desarrollado por Oracle Corporation como parte de su familia de productos de virtualización. Por medio de esta aplicación es posible instalar sistemas operativos adicionales, conocidos como «*sistemas invitados*», dentro de otro sistema operativo «anfitrión», cada uno con su propio ambiente virtual. Entre los sistemas operativos soportados (en modo anfitrión) se encuentran GNU/Linux, Mac OS X, OS/2 Warp , Microsoft Windows, y Solaris/OpenSolaris, y dentro de ellos es posible virtualizar los sistemas operativos FreeBSD, GNU/Linux, OpenBSD, OS/2 Warp, Windows, Solaris, MS-DOS y muchos otros.

ANEXOS

MANUAL DE INSTALACION

BIENVENIDO, al manual de Instalacion de GlusterFa Cliente y Servidor:

En esta práctica, configuraremos el volumen replicado de GlusterFS en Ubuntu 16.04 y Ubuntu Server 16.04 montado en máquinas virtuales desde VirtualBox. Esta guía también debería funcionar en versiones anteriores de Ubuntu como Ubuntu 14.04 / Ubuntu 12.04.

El Volumen replicado de Glusterfs es como un **RAID 1**, y el volumen mantiene copias exactas de los datos en todos los **bricks**. Puede decidir el número de réplicas mientras crea el volumen, por lo que necesitará tener al menos dos bricks para crear un volumen con dos réplicas o tres bricks para crear un volumen de 3 réplicas.

Este volumen proporciona una mejor confiabilidad de datos y redundancia de datos.

El siguiente manual de instalación esta creado para el siguiente escenario.

Requisitos:

Aquí, vamos a configurar el volumen de GlusterFS con dos réplicas. Asegúrese de tener dos sistemas de 64 bits (ya sean virtuales o físicos) puede instalarlos sobre VirtualBox por lo menos con 1 GB de memoria y un disco duro de repuesto en cada sistema de 10 GB

Host Name	IP Address	OS	Memory	Disk	Purpose
gluster1	192.168.2.5	Ubuntu Server 16.04 LTS	1GB	/dev/sdb (5GB)	Storage Node 1
gluster2	192.168.2.6	Ubuntu Server 16.04 LTS	1GB	/dev/sdb (5GB)	Storage Node 2
client1	192.168.2.7	Ubuntu 16.06 LTS Desktop	NA	NA	Cliente

Tabla de asignacion de Host Name, IP Address y características de los equipos virtuales.

Debe Asignar las IP estáticas en modo **usuario root**, escribir las siguientes instrucciones, en la terminal de cada equipo servidor y editar con las lineas siguientes.

```
nano /etc/network/interfaces
```

```
iface emp0s3
#iface emp0s3 inet dhcp
iface emp0s3 inet static
address 192.168.2.5 * para server1 192.168.2.6 para server2
netmask 255.255.255.0
```

Agregar Glusterfs-Server

En máquina gluster1 y gluster2 con Sistema operativo Ubuntu Server 16.04 / 14.04 LTS:

Descargar repositorios con la siguiente instrucciones desde terminal

```
sudo apt-get install -y software-properties-common  
  
sudo add-apt-repository ppa:gluster/glusterfs-3.8
```

INSTALAR GLUSTERFS-SERVER

Una vez que haya agregado el repositorio en sus sistemas, estamos listos para instalar GlusterFS.

Actualiza el caché del repositorio.

```
sudo apt-get update
```

Instalar el paquete de glusterfs server en ambos servidores

```
sudo apt-get install -y glusterfs-server
```

Iniciar el servicio glusterfs-server en ambos nodos gluster1 y gluster2

```
sudo service glusterfs-server start
```

Verificar que el servicio de router está corriendo en ambos gluster1 y gluster2

```
raj@gluster1:~$ sudo service glusterfs-server status  
  
• glusterfs-server.service – LSB: GlusterFS server  
  
   Loaded: loaded (/etc/init.d/glusterfs-server; bad; vendor preset: enabled)  
  
   Active: active (running) since Sat 2016-09-24 21:47:20 IST; 1min 6s ago  
  
     Docs: man:systemd-sysv-generator(8)  
  
  Process: 1570 ExecStop=/etc/init.d/glusterfs-server stop (code=exited, status=0/SUCCESS)
```

```
Process: 1664 ExecStart=/etc/init.d/glusterfs-server start (code=exited,
status=0/SUCCESS)
```

```
Tasks: 7 (limit: 512)
```

```
Memory: 25.5M
```

```
CPU: 3.552s
```

```
CGroup: /system.slice/glusterfs-server.service
```

```
└─1673 /usr/sbin/glusterd -p /var/run/glusterd.pid
```

```
Feb 21 21:47:16 192.168.2.5 systemd[1]: Starting LSB: GlusterFS server...
```

```
Feb 21 21:47:16 192.168.2.6 glusterfs-server[1664]: * Starting glusterd service glusterd
```

```
Feb 21 21:47:20 192.168.2.5 glusterfs-server[1664]: ...done.
```

```
Feb 21 21:47:20 192.168.2.6 systemd[1]: Started LSB: GlusterFS server.
```

CONFIGURAR EL FIREWALL

Debemos deshabilitar el firewall o configurar el firewall para permitir todas las conexiones dentro de un clúster., hacer este procedimiento.

En Gluster 1, IP-Addres 192.168.2.5

```
sudo iptables -I INPUT -p all -s 192.168.2.5 -j ACCEPT
```

En Gluster2, IP-Addres 192.168.2.6

```
sudo iptables -I INPUT -p all -s 192.168.2.6 -j ACCEPT
```

Aquí se ejecutan todos los comandos de GlusterFS en cada nodo para verificar la conexión entre ellos. Desde Glúster 1 hacia glúster 2

```
raj@gluster1:~$ sudo gluster peer probe 192.168.2.6  
peer probe: success.
```

Verificar la disponibilidad de cada nodo

```
raj@gluster1:~$ sudo gluster peer status  
  
Number of Peers: 1  
  
Hostname: 192.168.2.6  
  
Uuid: 51470928-dfa8-42e1-a221-d7bbcb8c13bd  
  
State: Peer in Cluster (Connected)
```

Listar el grupo de almacenamiento.

```
raj@gluster1:~$ sudo gluster pool list
```

UUID	Hostname	State
51470928-dfa8-42e1-a221-d7bbcb8c13bd	192.168.2.5	Connected
dc7c1639-d21c-4adf-b28f-5150229e6980	localhost	Connected

Verificamos que ambos nodos están a disponibilidad del gluster.

Crear un brick (directory) llamado “**gvol0**” En los archivos del sistema, esto en cada nodo

```
sudo mkdir -p /data/gluster/gvol0
```

Como vamos a usar el volumen replicado, debemos crear el volumen llamado **"gvol0"** con dos réplicas.es aquí donde se enlazan las replicas.

```
raj@gluster1:~$ sudo gluster volume create gvol0 replica 2 192.168.2.5:/data/gluster/gvol0
192.168.2.6:/data/gluster/gvol0 force

volume create: gvol0: success: please start the volume to access data
```

Iniciar el volumen creado

```
raj@gluster1:~$ sudo gluster volume start gvol0

volume start: gvol0: success
```

Verificar el estado del funcionamiento del volumen creado

```
raj@gluster1:~$ sudo gluster volume info gvol0

Volume Name: gvol0
Type: Replicate
Volume ID: ca102e4b-6cd1-4d9d-9c5a-03b882c76da0
Status: Started
Snapshot Count: 0
Number of Bricks: 1 x 2 = 2
Transport-type: tcp
Bricks:
Brick1:192.168.2.5:/data/gluster/gvol0
Brick2: 192.168.2.6:/data/gluster/gvol0
Options Reconfigured:
transport.address-family: inet
performance.readdir-ahead: on
```

```
nfs.disable: on
```

Configurar e instalar el Glusterfs-Cliente

Instale el paquete `glusterfs-client` para soportar el montaje de los sistemas de archivos GlusterFS. Ejecute todos los comandos como usuario `root`.

Descargar repositorios con la siguiente instrucciones desde terminal

```
sudo apt-get install -y software-properties-common  
  
sudo add-apt-repository ppa:gluster/glusterfs-3.8
```

Instalar `gluster-client`

```
### Ubuntu ###  
  
apt-get install -y glusterfs-client
```

Crea un directorio para montar el sistema de archivos GlusterFS.

```
mkdir -p /var/www/html
```

Ahora como usuario `root`, montaremos el sistema de archivos GlusterFS en `/var/www/html` usando el siguiente comando. para ambas direcciones IP

```
mount -t glusterfs 192.168.2.5:/gvol0 /var/www/html  
  
mount -t glusterfs 192.168.2.6:/gvol0 /var/www/html
```

Verificar el sistema de archivos GlusterFS montado.

```
[root@client ~]# df -hP /var/www/html  
  
Filesystem                Size  Used Avail Use% Mounted on  
192.168.2.5:/gvol0 4.8G   11M  4.6G   1% /mnt/glusterfs
```

También puede usar el comando below para verificar el sistema de archivos GlusterFS.

```
[root@client ~]# cat /proc/mounts

rootfs / rootfs rw 0 0

sysfs /sys sysfs rw,seclabel,nosuid,nodev,noexec,relatime 0 0

proc /proc proc rw,nosuid,nodev,noexec,relatime 0 0

devtmpfs /dev devtmpfs rw,seclabel,nosuid,size=490448k,nr_inodes=122612,mode=755 0 0

securityfs /sys/kernel/security securityfs rw,nosuid,nodev,noexec,relatime 0 0

tmpfs /dev/shm tmpfs rw,seclabel,nosuid,nodev 0 0

devpts /dev/pts devpts rw,seclabel,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000 0 0

tmpfs /run tmpfs rw,seclabel,nosuid,nodev,mode=755 0 0

tmpfs /sys/fs/cgroup tmpfs ro,seclabel,nosuid,nodev,noexec,mode=755 0 0

cgroup /sys/fs/cgroup/systemd cgroup
rw,nosuid,nodev,noexec,relatime,xattr,release_agent=/usr/lib/systemd/systemd-cgroups-
agent,name=systemd 0 0

pstore /sys/fs/pstore pstore rw,nosuid,nodev,noexec,relatime 0 0

cgroup /sys/fs/cgroup/memory cgroup rw,nosuid,nodev,noexec,relatime,memory 0 0

cgroup /sys/fs/cgroup/perf_event cgroup rw,nosuid,nodev,noexec,relatime,perf_event 0 0

cgroup /sys/fs/cgroup/cpuset cgroup rw,nosuid,nodev,noexec,relatime,cpuset 0 0

cgroup /sys/fs/cgroup/devices cgroup rw,nosuid,nodev,noexec,relatime,devices 0 0

cgroup /sys/fs/cgroup/freezer cgroup rw,nosuid,nodev,noexec,relatime,freezer 0 0

cgroup /sys/fs/cgroup/cpu,cpuacct cgroup rw,nosuid,nodev,noexec,relatime,cpuacct,cpu 0 0

cgroup /sys/fs/cgroup/blkio cgroup rw,nosuid,nodev,noexec,relatime,blkio 0 0

cgroup /sys/fs/cgroup/hugetlb cgroup rw,nosuid,nodev,noexec,relatime,hugetlb 0 0
```



```

cgroup /sys/fs/cgroup/net_cls cgroup rw,nosuid,nodev,noexec,relatime,net_cls 0 0

configfs /sys/kernel/config configfs rw,relatime 0 0

/dev/mapper/centos-root / xfs rw,seclabel,relatime,attr2,inode64,noquota 0 0

selinuxfs /sys/fs/selinux selinuxfs rw,relatime 0 0

systemd-1 /proc/sys/fs/binfmt_misc autofs
rw,relatime,fd=25,pgrp=1,timeout=300,minproto=5,maxproto=5,direct 0 0

hugetlbfs /dev/hugepages hugetlbfs rw,seclabel,relatime 0 0

debugfs /sys/kernel/debug debugfs rw,relatime 0 0

mqueue /dev/mqueue mqueue rw,seclabel,relatime 0 0

/dev/mapper/centos-home /home xfs rw,seclabel,relatime,attr2,inode64,noquota 0 0

/dev/sda1 /boot xfs rw,seclabel,relatime,attr2,inode64,noquota 0 0

tmpfs /run/user/0 tmpfs rw,seclabel,nosuid,nodev,relatime,size=100136k,mode=700 0 0

gluster1.itzgeek.local:/gvol0 /mnt/glusterfs fuse.glusterfs
rw,relatime,user_id=0,group_id=0,default_permissions,allow_other,max_read=131072 0 0

fusectl /sys/fs/fuse/connections fusectl rw,relatime 0 0

```

GlusterFS Lado del Servidor:

Para verificar la replicación, monte el volumen GlusterFS creado en el mismo nodo de almacenamiento.

```

raj@gluster1:~$ sudo mount -t glusterfs 192.168.2.5:/gvol0 /mnt

raj@gluster2:~$ sudo mount -t glusterfs 192.168.2.6:/gvol0 /mnt

```

Los datos dentro del directorio **/mnt** de ambos nodos siempre serán los mismos (replicación).

GlusterFS Lado del Cliente

Vamos a crear archivos dentro de los directorios creador

```
touch /var/www/html/index.php  
touch /var/www/html/index1.php
```

Verificar la creación de los archivos

```
[root@client ~]# ll /var/www/html  
  
total 0  
  
-rw-r--r--. 1 root root 0 Feb 21 11:22 index.php  
-rw-r--r--. 1 root root 0 Feb 21 11:22 index1.php
```

Revisar que los archivos han sido duplicados dentro del /mnt.

```
raj@gluster1:~$ ll /mnt/  
  
total 12  
  
drwxr-xr-x  4 root root 4096 Feb 21 20:52 ./  
drwxr-xr-x 24 root root 4096 Feb 21 21:57 ../  
  
-rw-r--r--  1 root root    0 Feb 21 20:52 index.php  
-rw-r--r--  1 root root    0 Feb 21 20:52 index1.php  
  
raj@gluster2:~$ ls -al /mnt/  
  
total 12  
  
drwxr-xr-x  4 root root 4096 Feb 21 06:30 .  
drwxr-xr-x 23 root root 4096 Feb 21 08:39 ..
```

```
-rw-r--r-- 1 root root 0 Feb 21 2018 index.php
-rw-r--r-- 1 root root 0 Feb 21 2018 index1.php
```

Y finalmente se concluye con la instalación y configuración de
GlusterFS Server y Cliente.