

TERCER-EXAMEN-PRACTICA-SO-RESUEL...



sky_shadow



Sistemas Operativos



2º Grado en Ingeniería Informática - Tecnologías Informáticas



Escuela Técnica Superior de Ingeniería Informática
Universidad de Sevilla



[Accede al documento original](#)



Escuela de
Organización
Industrial

Contigo que evolucionas.
Contigo que lideras. Contigo que transformas.

Esto es EOI.
Mismo propósito,
nueva energía.



Descubre más aquí



EOI Escuela de
Organización
Industrial

Importante

Puedo eliminar la publi de este documento con 1 coin

¿Cómo consigo coins? → Plan Turbo: barato
→ Planes pro: más coins

perdo
espacio



TERCER EXAMEN PRÁCTICA SO 2024 - Grupos L1, L2 y L3

1. Vamos a preparar las tres máquinas **server1**, **server2** y **server3** para convertirlas en nodos GlusterFS. Para ello, en cada una de las tres máquinas instale y ponga en marcha el servicio de lado servidor de GlusterFS, para que a partir de ahora el cada servidor se convierta en un nodo GlusterFS. Puede copiar y pegar los comandos entre un servidor y otro. Responda cada comando una vez, basta con que indique en qué máquinas ejecuta cada comando

```
sudo apt update

sudo apt install glusterfs-server

sudo systemctl start glusterd

sudo systemctl enable glusterd
```

2. Desde **server1**, cree un TSP compuesto por **server1**, **server2** y **server3**. Para comprobar que todo ha ido en orden, liste a continuación los nodos del TSP y obtenga además el estado de los demás nodos del TSP (todo ello desde **server1**)

```
sudo gluster peer probe server2

sudo gluster peer probe server3

sudo gluster pool list

sudo gluster peer status
```

3. Cree a continuación un volumen GlusterFS de tipo distribuido llamado **userSpace** usando como bricks los sistemas de archivos montados en **/gs/br1** de **server1** y **server2**, y otro volumen de tipo replicado llamado **dataSpace** formado por los sistemas de archivos montados en **/gs/br2** de **server1**, **server2** y **server3**. Obtenga la lista de volúmenes existentes, y active ambos volúmenes.

```
sudo gluster volume create userSpace server1:/gs/br1/brick
server2:/gs/br1/brick

sudo gluster volume create dataSpace replica 3 server1:/gs/br2/brick
server2:/gs/br2/brick server3:/gs/br2/brick
```

Necesito
concentración

ali ali ooh
esto con 1 coin me
lo quito yo...

WUOLAH

WUOLAH

```
sudo gluster volume list

sudo gluster volume start userSpace

sudo gluster volume start dataSpace
```

4. En **server4**, cree sendos directorios llamados **/data** y **/user**, y monte los volúmenes **dataSpace** y **userSpace** respectivamente en ellos, cambiando el usuario y grupo de ambos a **administrador** y **administrador**, respectivamente. Instale previamente el software que pueda necesitar.

A continuación, ejecute el siguiente comando para crear archivos de prueba en **/user** (cópielo y péguelo para mayor seguridad):

for i in \$(seq -w 0 399); do echo "Este es el archivo de prueba \$i" > /user/archivo-\$i.txt; done

```
sudo apt update

sudo apt install glusterfs-client

sudo mkdir /data /user

sudo mount -t glusterfs -o _netdev server1:/userSpace /user

sudo mount -t glusterfs -o _netdev server1:/dataSpace /data

sudo chown -R administrador /user /data

sudo chgrp -R administrador /user /data

for i in $(seq -w 0 399); do echo "Este es el archivo de prueba $i" >
/user/archivo-$i.txt; done
```

5. Ahora estará esperando que añadamos el brick que falta al volumen **userSpace**, o que apaguemos un servidor para simular la pérdida de un brick en el volumen replicado **dataSpace**, ¿verdad? ¡Pues no! Vamos a escalar hacia abajo el volumen **userSpace**. Quite el brick aportado por **server2** a dicho volumen.

```
sudo gluster volume remove-brick userSpace server2:/gs/br2/brick start

sudo gluster volume remove-brick userSpace server2:/gs/br2/brick status

sudo gluster volume remove-brick userSpace server2:/gs/br2/brick commit
```

6. Vamos a desplegar una aplicación escrita en JavaScript. Podrá encontrar los archivos de la aplicación en el directorio **/home/ssoo-shared/** de su servidor, en un archivo llamado **calc.tgz**. Descargue por sftp dicho archivo y extraiga su contenido (no hace falta que responda los comandos a utilizar para ello).

Para desplegar esta aplicación junto con su entorno, ha de seguirse los siguientes pasos:

- Tomaremos como punto de partida una imagen con la versión 2 de apache (imagen **httpd:2** en registro central de Docker)
- Sobre esta imagen, instalaremos la herramienta **flex** (contenida en el paquete con el mismo nombre)
- Copiaremos el archivo **calc.html** al directorio **/usr/local/apache2/htdocs/** de la imagen
- Crearemos en la imagen un directorio **/lib/flexlib**, en el cual copiaremos los archivos **flexlib0.d**, **flexlib1.d**, **flexlib2.d**
- La aplicación debe encontrar una variable de entorno **FLEXMODE** que por defecto tendrá como valor la cadena "strict"
- El comando de arranque de la aplicación será el declarado en la imagen en la cual se basa, con lo que no hay que cambiarlo.

Escriba el **Dockerfile** y el comando Podman con el que se genere una imagen llamada **calc:1.0**. Reetiquete la imagen anteponiendo su nombre de usuario en el registro central de docker, y suba la imagen a dicho registro central.

Nota para despistad@s: para descargar el archivo por sftp:

```
sftp mi-cuenta@miservidor
cd /home/ssoo-shared
get calc.tgz
```

```
mkdir calc

cd calc

sftp <uvus>@<servidor>.lsi.us.es

cd /home/ssoo-shared

get calc.tgz

exit

tar -xzf archivos.tgz

podman image pull httpd:2

nano Dockerfile
```

Importante

Puedo eliminar la publi de este documento con 1 coin

¿Cómo consigo coins? → Plan Turbo: barato
→ Planes pro: más coins

pierdo espacio



Necesito concentración

ali ali oohh
esto con 1 coin me
lo quito yo...

WUOLAH

```
FROM httpd:2
RUN apt update
RUN apt install -y flex
COPY calc.html /usr/local/apache2/htdocs/
RUN mkdir -p /lib/flexlib
COPY ["flexlib0.d", "flexlib1.d", "flexlib2.d", "/lib/flexlib"]
ENV FLEXMODE="strict"

podman build -t calc:1.0 .

podman image tag calc:1.0 <usuario>/calc:1.0

podman login

podman image push <usuario>/calc:1.0
```

7. Ejecute la imagen anteriormente creada usando **"jscalc"** como nombre de contenedor, de forma que la ejecución del mismo se lleve a cabo en segundo plano y mapeando el puerto **80** del contenedor sobre el puerto **4000** del anfitrión, redefiniendo el valor de la variable **FLEXMODE** a "relaxed".

Puede comprobar que la aplicación se esté ejecutando abriendo la URL <http://localhost:4000/calc.html> en el navegador del anfitrión.

A continuación, renombre el contenedor que acaba de crear como **"calculadora"**. Acto seguido, detenga la ejecución del contenedor y póngalo de marcha de nuevo. Finalmente, detenga otra vez el contenedor, y elimínelo. Elimine también ambas imágenes creadas de su registro local.

DISCLAIMER: aunque la aplicación se ejecute aparentemente bien (se trata de una calculadora), eso no significa que ni esta cuestión ni la anterior estén perfectas... aunque siempre es un buen indicio

```
podman run --name jscalc -d -p 4000:80 -e FLEXMODE="relaxed"
<usuario>/calc:1.0

podman container rename jscalc calculadora

podman container kill calculadora

podman container start calculadora

podman container kill calculadora
```

WUOLAH

```
podman container rm calculadora
```

```
podman image rm -a
```