

Universidad Icesi
Christian Cárdenas A00212740
Luis F. Rosales A00315320
Sistemas distribuidos
Prof. Daniel Barragán
Examen 3
Repositorio: www.github.com/cdcardenas/sd-exam3

Resumen: Para el desarrollo de este trabajo se utilizaran 4 máquinas físicas en modo swarm(1 maestro y 3 trabajadores). En cada máquina se instalará GlusterFS para crear un clúster almacenamiento compartido entre las cuatro máquinas

Maquina maestra: en esta máquina se ejecutaron los siguientes comandos, que tienen como finalidad la instalación y configuración de glusterfs.

- se crea y se añade un punto de montaje de almacenamiento para glusterfs.
- se crea un volumen en gluster
- se da los permisos al host maestro y se inicializa el volumen
- se agrega el nuevo volumen al archivo fstab que administra los filesystems estáticos
- se monta el nuevo volumen

```
sudo apt-get install glusterfs-server
sudo mkdir -p /var/gluster/volumes /var/volumes
sudo gluster volume create volumes
$(hostname):/var/gluster/volumes force
sudo gluster volume set volumes auth.allow 127.0.0.1
sudo gluster volume start volumes
sudo tee -a /etc/fstab <<EOF
localhost:/volumes /var/volumes glusterfs
defaults,_netdev,noauto,x-systemd.automount,noatime,nodiratime 0 0

EOF

sudo mount /var/volumes
```

Fig. 1: Instalación GlusterFS

A continuación, iniciamos el cluster de docker con docker swarm

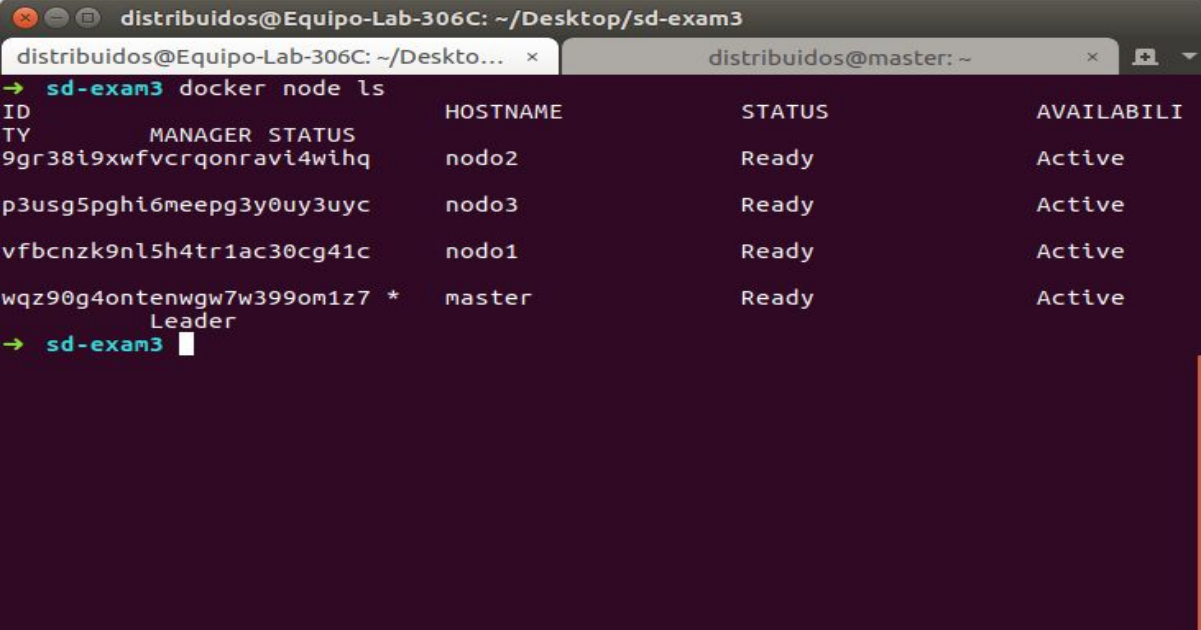
```
docker swarm init --advertise-addr 192.168.130.128
```

Fig. 2: Inicialización Docker Swarm

Iniciamos el cluster de docker y ya se pueden agregar los nodos con el comando:

```
docker swarm join --token
SWMTKN-1-3l4gq7xd61q948um9voc83u8vt0b1rr1kg7uctp9rpm0exq0ty-bf5phwb5iya0
izqjqynl6q9o2 192.168.130.128:2377
```

Fig. 3: Comando para unirse al cluster



```
distribuidos@Equipo-Lab-306C: ~/Desktop/sd-exam3
→ sd-exam3 docker node ls
ID                MANAGER STATUS    HOSTNAME    AVAILABILITY
9gr38i9xwfvcrqonravi4wihq      nodo2    Ready     Active
p3usg5pghi6meepg3y0uy3uyc      nodo3    Ready     Active
vfbcnz9nl5h4tr1ac30cg41c      nodo1    Ready     Active
wqz90g4ontenw7w399om1z7 *    master    Ready     Active
                                Leader
→ sd-exam3
```

Fig. 4: Lista de nodos cluster Swarm

Ahora necesitamos añadir los nodos al cluster de glusterfs, para ello primero debemos crear los puntos de montura en cada nodo de red y luego agregarlos desde el nodo maestro

En cada nodo worker ejecutamos:

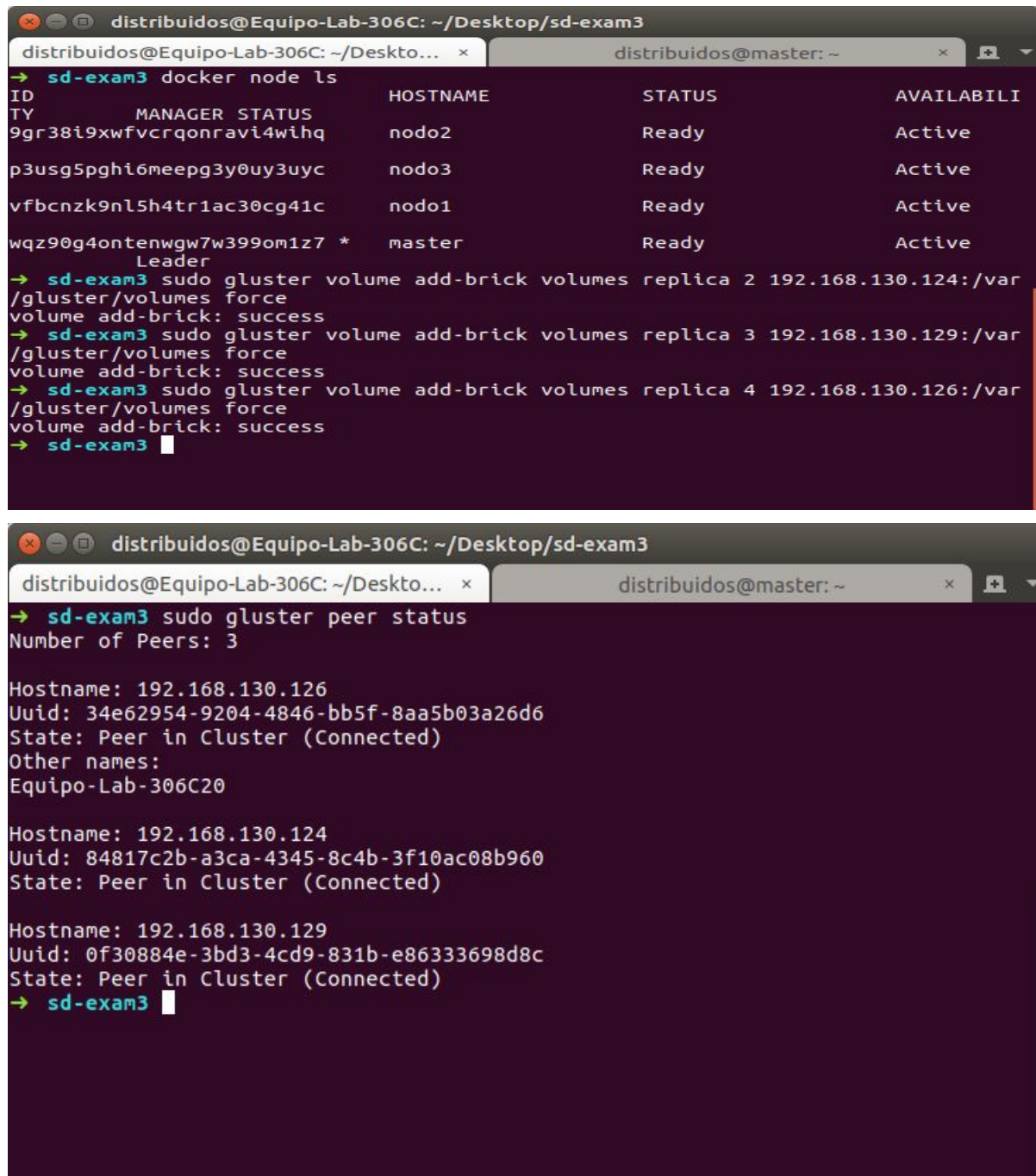
```
sudo mkdir -p /var/gluster/volumes /var/volumes
sudo tee -a /etc/fstab <<EOF
localhost:/volumes /var/volumes glusterfs
defaults,_netdev,noauto,x-systemd.automount 0 0
EOF
```

Fig. 5: Configuración GlusterFS en nodos de red

Desde el nodo maestro ejecutamos:

```
sudo gluster peer probe 192.168.130.126
sudo gluster volume add-brick volumes replica 2
192.168.130.126/var/gluster/volumes force
#ejecutar en cada nodo cambiando únicamente la dirección ip
```

Fig. 6: Agregando nodos de red al cluster de almacenamiento



```
distribuidos@Equipo-Lab-306C: ~/Desktop/sd-exam3
distribuidos@Equipo-Lab-306C: ~/Desкто... x distribuidos@master: ~ x
→ sd-exam3 docker node ls
ID                                     HOSTNAME    STATUS    AVAILABILITY
TY      MANAGER STATUS
9gr38i9xwfvcrqonravi4wihq          nodo2      Ready     Active
p3usg5pghi6meepg3y0uy3uyc          nodo3      Ready     Active
vfbcnz9nl5h4tr1ac30cg41c          nodo1      Ready     Active
wqz90g4ontenwgw7w399om1z7 *      master     Ready     Active
Leader
→ sd-exam3 sudo gluster volume add-brick volumes replica 2 192.168.130.124:/var
/gluster/volumes force
volume add-brick: success
→ sd-exam3 sudo gluster volume add-brick volumes replica 3 192.168.130.129:/var
/gluster/volumes force
volume add-brick: success
→ sd-exam3 sudo gluster volume add-brick volumes replica 4 192.168.130.126:/var
/gluster/volumes force
volume add-brick: success
→ sd-exam3 █

distribuidos@Equipo-Lab-306C: ~/Desktop/sd-exam3
distribuidos@Equipo-Lab-306C: ~/Desкто... x distribuidos@master: ~ x
→ sd-exam3 sudo gluster peer status
Number of Peers: 3

Hostname: 192.168.130.126
Uuid: 34e62954-9204-4846-bb5f-8aa5b03a26d6
State: Peer in Cluster (Connected)
Other names:
Equipo-Lab-306C20

Hostname: 192.168.130.124
Uuid: 84817c2b-a3ca-4345-8c4b-3f10ac08b960
State: Peer in Cluster (Connected)

Hostname: 192.168.130.129
Uuid: 0f30884e-3bd3-4cd9-831b-e86333698d8c
State: Peer in Cluster (Connected)
→ sd-exam3 █
```

Fig. 7: cluster de docker y cluster de almacenamiento

Para desplegar el cluster configuramos el archivo docker-compose.yml, para ello utilizamos de plantilla el archivo del sd-exam2 y agregamos una imagen para el servicio de base de datos

```
version: "3.3"

services:

  mysql:
    image: mysql:latest
    networks:
      - net
    volumes:
      - type: bind
        source: /var/volumes/mrw-cloud/mysql
        target: /var/lib/mysql
    environment:
      - 'MYSQL_DATABASE=nextcloud'
      - 'MYSQL_USER=nextcloud'
      - 'MYSQL_PASSWORD=SomeSecretPassword'
      - 'MYSQL_ROOT_PASSWORD=SomeSecretRootPassword'
  nextcloud:
    image: mwaeckerlin/nextcloud:latest
    networks:
      - net
    ports:
      - 8001:80
    environment:
      - 'MYSQL_PASSWORD=SomeSecretPassword'
      - 'MAX_INPUT_TIME=7200'
      - 'URL=mrw.cloud'
      - 'ADMIN_PWD=GuiAdministrationPassword'
    volumes:
      - type: bind
        source: /var/volumes/mrw-cloud/config
        target: /var/www/nextcloud/config
      - type: bind
        source: /var/volumes/mrw-cloud/data
        target: /var/www/nextcloud/data
```

```
whoami:
```

```
image: tutum/hello-world
networks:
- net
ports:
- "80:80" # puerto de funcionamiento
logging:
driver: "fluentd" # Logging Driver
options:
```

```
tag: tutum # TAG
# fluentd-address: 127.0.0.1:24224
deploy:
resources:
limits:
  cpus: '0.10'
  memory: 20M
reservations:
  cpus: '0.05'
  memory: 10M

restart_policy:
  condition: on-failure
  delay: 20s
  max_attempts: 3
  window: 120s
mode: replicated
replicas: 3

placement:
constraints: [node.role == worker] # Se ejecutara en los dos nodos
workers
update_config:
  delay: 2s

vizualizer:
image: dockersamples/visualizer
volumes:
- /var/run/docker.sock:/var/run/docker.sock
ports:
- "8080:8080"
networks:
- net
logging:
driver: "fluentd"
```

```
options:
tag: visualizer #TAG
deploy:
  restart_policy:
    condition: on-failure
    delay: 20s
    max_attempts: 3
    window: 120s
    mode: replicated # un contenedor por nodo manager
```

```
replicas: 1
  update_config:
    delay: 2s
  placement:
    constraints: [node.role == manager]

fluentd:
  image: cdcardenas/myfluentd:v0.1.0
  volumes:
    #- ./Logs:/fluentd/log
    - ./fluentd/conf:/fluentd/etc
  ports:
    - "24224:24224"
    - "24224:24224/udp"
  networks:
    - net
  deploy:
    restart_policy:
      condition: on-failure
      delay: 20s
      max_attempts: 3
      window: 120s
    mode: replicated
  replicas: 1
  placement:
    constraints: [node.role == manager]
  update_config:
    delay: 2s

elasticsearch:
  image: elasticsearch #Elastic Search
  ports:
    - "9200:9200"
  networks:
    - net
```

```
environment:
  - bootstrap.memory_lock=true
  - "ES_JAVA_OPTS=-Xms512m -Xmx512m"
logging:
driver: "json-file"
options:
  max-size: 10M
  max-file: 1
deploy:
restart_policy:
condition: on-failure
delay: 20s
max_attempts: 3
window: 120s
mode: replicated
replicas: 1
placement:
constraints: [node.role == manager]
update_config:
delay: 2s
resources:
limits:
  memory: 1000M
volumes:
  - ./esdata:/usr/share/elasticsearch/data
```

```
kibana:
  image: kibana
  ports:
    - "5601:5601" #puerto de funcionamiento
  networks:
    - net
  logging:
driver: "json-file"
options:
  max-size: 10M
  max-file: 1
deploy:
restart_policy:
condition: on-failure
delay: 20s
max_attempts: 3
window: 120s
mode: replicated
replicas: 1
```

```

    placement:
      constraints: [node.role == manager] #un contenedor por nodo
manager
    update_config:
      delay: 2s

networks:
  net:

```

Fig.8: Archivo Docker compose con el servicio de base de datos

```

→ sd-exam2 docker service ls
ID                NAME                PORTS                MODE                REPLICAS
IMAGE
a7cs1o5csy51      test_whoami          *:80->80/tcp          replicated           3/3
tutum/hello-world:latest
h70xi1bx9c5a      test_elasticsearch   *:9200->9200/tcp      replicated           1/1
elasticsearch:latest
ihgdg3scxzvn      test_kibana          *:5601->5601/tcp      replicated           1/1
kibana:latest
luevg9e0etjy      test_mysql           replicated            1/1
mysql:latest
nhrrq1lu39d7      test_nextcloud       *:8001->80/tcp        replicated           0/1
mwaeckerlin/nextcloud:latest
ty37oiy8ld84      test_fluentd         *:24224->24224/tcp,*:24224->24224/udp replicated           1/1
cdcardenas/myfluentd:v0.1.0
wcyR5u5qns5z      test_vizualizer      *:8080->8080/tcp      replicated           1/1
dockersamples/visualizer:latest
→ sd-exam2

```

Fig. 9: Servicios docker swarm

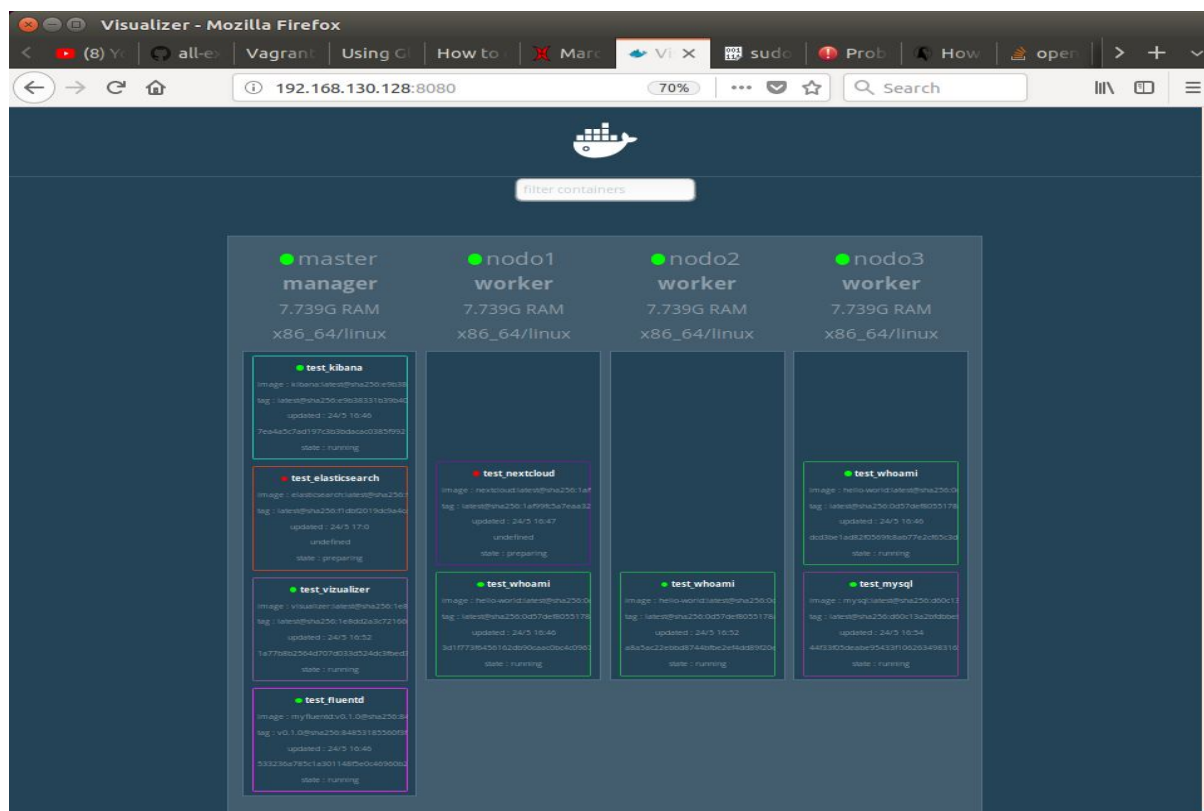


Fig. 10: servicios y nodos

Ahora verificamos que el glusterfs esté replicando la información en los demás nodos, para esto podemos acceder vía SSH o revisar manualmente en cada nodo si el la dirección `~/var/volumes` está almacenando la información.

```
distribuidos@nodo1: ~
→ ~ sudo ssh distribuidos@192.168.130.126
[sudo] password for distribuidos:
The authenticity of host '192.168.130.126 (192.168.130.126)' can't be established.
ECDSA key fingerprint is SHA256:isbukd+yqCM4ZYwPq80eazEp022b4c5XqIbduazQET8.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.130.126' (ECDSA) to the list of known hosts.
distribuidos@192.168.130.126's password:
Welcome to Ubuntu 16.10 (GNU/Linux 4.8.0-59-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

133 packages can be updated.
0 updates are security updates.

Your Ubuntu release is not supported anymore.
For upgrade information, please visit:
http://www.ubuntu.com/releaseendoflife

New release '17.10' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Fri Dec 15 15:49:04 2017 from 192.168.130.5
distribuidos@nodo1:~$
```

```
distribuidos@nodo1: /var/volumes/mrw-cloud
distribuidos@Equipo-Lab-306C: ~/Deskto... x distribuidos@nodo1: /var/volumes/mrw-... x
Last login: Fri Dec 15 15:49:04 2017 from 192.168.130.5
distribuidos@nodo1:~$ cd /var/volumes/
distribuidos@nodo1:/var/volumes$ ls
mrw-cloud
distribuidos@nodo1:/var/volumes$ cd mrw-cloud/
distribuidos@nodo1:/var/volumes/mrw-cloud$ ls
config data mysql
distribuidos@nodo1:/var/volumes/mrw-cloud$ tree
.
├── config
│   └── config.php
├── data
│   ├── index.html
│   └── nextcloud.log
└── mysql
    ├── auto.cnf
    ├── ca-key.pem
    ├── ca.pem
    ├── client-cert.pem
    ├── client-key.pem
    ├── ibdata1
    ├── ib_logfile0
    ├── ib_logfile1
    ├── ibtmp1
    └── mysql
        ├── general_log_195.sdi
        ├── general_log.CSM
        ├── general_log.CSV
        ├── slow_log_196.sdi
        ├── slow_log.CSM
        ├── slow_log.CSV
        └── mysql.ibd
```

Como se puede observar la información se está replicando correctamente poco tiempo después de haber sido almacenada en el punto de montura inicial. De esta forma, una vez bajen los servicios, la información persistirá y se podrá acceder a ella desde cualquier nodo de red en la dirección **/var/volumes**

Problemas: Durante la ejecución del parcial surgieron varios problemas de incompatibilidad de con Docker compose, estos se solucionan utilizando la versión 3.3 del formato para yaml. un segundo problema fue tuvo utilizarse como almacenamiento carpetas y no discos duros, para esto se utilizaron puntos de montura en los distintos nodos de red. Otro problema que retrasó el desarrollo del examen fue que en todas las máquinas de la sala de redes el gestor de paquetes apt no está funcionando correctamente, por lo que no se podían instalar software como glusterfs-server o xfsprogs. El problema se soluciono eliminando los archivos sources.list.* de cada máquina y reemplazandolos con la versión por defecto de la distribución Xenial de ubuntu 16.04 con lo cual ya se pudo instalar correctamente el software.