glusterfs vagrant POC

vagrant

tests

- 3 nodos, cae 1
  - los otros dos continúan funcionando
  - al reincorporarse el tercero, restituye la información
- 2 nodos, añadir tercero posteriormente
  - `gluster peer probe node3`
  - `gluster volume add-brick gv0 replica 3 node3:/exports/sdb1/brick1`
  - `gluster volume status`
- 1 nodo, añadir segundo (pruebas processing 3)
  - no deja crear un volumen con replica si no hay nodos para soportarlo
  - no deja crear un volumen con replica 1
  - arbiter mode con replica 2?? → **NO**
  - node 1:

```
sudo gluster volume create gv0 node1:/exports/sdb1/brick1 # volume
create: gv0: success: please start the volume to access data
sudo gluster volume start gv0 # volume start: gv0: success
sudo mount -t glusterfs node1:/gv0 /mnt
```

  - node 2:

```
sudo gluster volume status
sudo mount -t glusterfs node2:/gv0 /mnt
```

  - node 1:

```
sudo gluster peer probe node2 #peer probe: success
sudo gluster volume add-brick gv0 replica 2 node2:/exports/sdb1/brick1 #
volume add-brick: success
```

```
Replica 2 volumes are prone to split-brain. Use Arbiter or Replica 3 to
avoid this. See:
http://docs.gluster.org/en/latest/Administrator%20Guide/Split%20brain%20a
nd%20ways%20to%20deal%20with%20it/.
```

install

- require: `vagrant plugin install vagrant-hosts`

Vagrantfile

```
VAGRANTFILE_API_VERSION = "2"

cluster = {
  "node1" => { :ip => "10.0.0.10", :cpus => 1, :mem => 1024 },
  "node2" => { :ip => "10.0.0.20", :cpus => 1, :mem => 1024 },
  "node3" => { :ip => "10.0.0.30", :cpus => 1, :mem => 1024 }
}
```

```
Vagrant.configure(VAGRANTFILE_API_VERSION) do |config|

  cluster.each_with_index do |(hostname, info), index|

    config.vm.define hostname do |cfg|

      cfg.vm.provider :virtualbox do |vb, override|
        config.vm.box = "debian/buster64"
        override.vm.network :private_network, ip: "#{info[:ip]}"
        override.vm.hostname = hostname
        vb.name = hostname
        vb.customize ["modifyvm", :id, "--memory", info[:mem], "--cpus",
info[:cpus], "--hwvirtex", "on"]

        file_to_disk = 'data/' + hostname + '-disk.vdi'
        unless File.exist?(file_to_disk)
          vb.customize ['createhd',
                        '--filename', file_to_disk,
                        '--size', 5 * 1024]
        end
        vb.customize ['storageattach', :id,
                      '--storagectl', 'SATA Controller',
                      '--port', 1,
                      '--device', 0,
                      '--type', 'hdd',
                      '--medium', file_to_disk]

      end # end provider

      config.vm.provision 'shell', path: './data/allnodes.sh'
      config.vm.provision :hosts, :sync_hosts => true

      node_script = './data/' + hostname + '.sh'
      if File.exists?(node_script) then
        config.vm.provision "file", source: node_script, destination: "$HOME/"
      else
        config.vm.provision "shell", inline: "echo SCRIPT not found!"
      end

    end # end config

  end # end cluster
end
```

allnodes.sh

```
[[ ! -f /etc/provision_env_disk_added_date ]] && {

  echo 'type=83' | sudo sfdisk /dev/sdb
  sudo mkfs.ext4 /dev/sdb1
  sudo mkdir -p /exports/sdb1
  echo "/dev/sdb1      /exports/sdb1   ext4 user,auto,defaults     0       0"
```

```
| sudo tee --append /etc/fstab
  sudo date > /etc/provision_env_disk_added_date

}

mount /exports/sdb1
sudo mkdir /exports/sdb1/brick1

sudo apt-get -y install gnupg
wget -O - https://download.gluster.org/pub/gluster/glusterfs/7/rsa.pub | sudo
apt-key add -
# DEBID=$(grep 'VERSION_ID=' /etc/os-release | cut -d '=' -f 2 | tr -d '"')
# DEBVER=$(grep 'VERSION=' /etc/os-release | grep -Eo '[a-z]+')
# DEBARCH=$(dpkg --print-architecture)
# echo "deb
https://download.gluster.org/pub/gluster/glusterfs/LATEST/Debian/${DEBID}/${DE
BARCH}/apt ${DEBVER} main" | sudo tee /etc/apt/sources.list.d/gluster.list
echo "deb
https://download.gluster.org/pub/gluster/glusterfs/LATEST/Debian/10/amd64/apt
buster main" | sudo tee /etc/apt/sources.list.d/gluster.list

sudo apt-get -y update

sudo apt-get -y install glusterfs-server
sudo systemctl start glusterd
sudo systemctl enable glusterd
```

node1.sh

```
#!/bin/bash

sudo gluster peer probe node2
sudo gluster peer probe node3

sudo gluster volume create gv0 replica 3 \
  node1:/exports/sdb1/brick1 \
  node2:/exports/sdb1/brick1 \
  node3:/exports/sdb1/brick1

sudo gluster volume start gv0

sudo mount -t glusterfs node1:/gv0 /mnt
```

node{2,3}.sh

```
#! /bin/bash

MY_HOST=$(cat /etc/hostname)
sudo mount -t glusterfs ${MY_HOST}:/gv0 /mnt
```

- en lugar de **node1**, podría ser cualquier otro de los dos nodos