CTDB and GlusterFS

Jacob Shucart   jacob@gluster.com   **408-770-1504 (pst)**

Craig Carl   craig@gluster.com   **408-829-9953 (pst)**

When used with Gluster replica volumes CTDB provides for highly available CIFS and NFS exports. CTDB adds virtual IP addresses (VIPs) and a heartbeat service to each Gluster server. For Gluster volumes that are exported via CIFS CTDB also adds a locking mechanism.

The installation examples provided here are for a basic CTDB setup. CTDB can be also configured with advanced multi-tiered failover, more information about CTDB is here - http://ctdb.samba.org.

EC2 does not support VIPs so CTDB is not supported with the Gluster AMI.

Prerequisites

- Gluster VSA | SSA 3.2
    upgrade to Gluster 3.2:
    ```
    yum clean
    yum update -y
    gluster-app-migrate 3.2
    ```

- CTDB
    ```
    yum install ctdb
    ```

- A Gluster replicated volume for the CTDB lock file.
    The lock file is only required for CIFS.
    The best practice is to have a volume dedicated to the lock file.
    Mount the volume using the Gluster native client on each storage node.
        The sample config below assumes the volume has been mounted at /gluster/lock

- Samba installed and configured. (CIFS only)
    Gluster volume you are exporting with Samba must be mounted locally.
    Samba should be configured to re-export these mounts.

- Port 4379 open between the Gluster servers.

Sample configuration, CTDB across 4 Gluster servers

Setting up CTDB is fairly simple. There are three configuration files, their contents are identical across every node. You can create these files in the same Gluster volume you are using for the lock file and then create a soft link for /etc/sysconfig/ctdb and change the paths below as appropriate.

- Stop Samba

- Add the following lines to the [global] section of your Samba configuration
  ```
  clustering = yes
  idmap backend = tdb2
  private dir = /gluster/lock
  ```

- /etc/sysconfig/ctdb
  ```
  CTDB_RECOVERY_LOCK=/gluster/lock/lockfile     #CIFS only
  CTDB_PUBLIC_ADDRESSES=/etc/ctdb/public_addresses
  CTDB_MANAGES_SAMBA=yes    #CIFS only
  CTDB_NODES=/etc/ctdb/nodes
  ```

- /etc/ctdb/public_addresses
  List the VIPs that CTDB should create, you need one VIP for every Gluster server.
  Replace *eth0* with the interface you want CTDB to use.
  ```
  192.168.1.20/24 eth0
  192.168.1.21/24 eth0
  192.168.1.22/24 eth0
  192.168.1.23/24 eth0
  ```

- /etc/ctdb/nodes
  List the IPs on each physical interface.
  ```
  192.168.1.60
  192.168.1.61
  192.168.1.62
  192.168.1.63
  ```

Starting and Verifying Your Configuration

- Start CTDB and set it to start automatically on boot.
  ```
  service ctdb start
  chkconfig ctdb on
  ```

- When CTDB starts it will start Samba automatically.
  ```
  chkconfig smb off
  ```

- Verify that CTDB is running.
  ```
  ctdb status
  ctdb ip
  ctdb ping -n all
  ```

- Mount a Gluster volume using any one of the VIPs.
  Shutdown the Gluster server with the VIP you mounted with, there will be about a 5 second pause and then I/O will resume.

Load Balancing With CIFS and NFS

CTDB provides highly available NFS and CIFS services across Gluster replica servers, it does not load balance those connections across the cluster. To prevent the interface on any single Gluster server becoming saturated it's important to balance NFS and CIFS mounts across all the Gluster servers, i.e. with 100 NFS clients and 10 Gluster servers you only want 10 clients mounted to each server.

You can use any sort of IP-based load balancing to spread client connections across the storage nodes. A round-robin DNS or WINS record that lists all VIPs is an easy way to do this. A RRDNS entry looks like this -

```
; zone file fragment

    gluster 1 IN A 192.168.1.21
    gluster 1 IN A 192.168.1.22
    gluster 1 IN A 192.168.1.23
    gluster 1 IN A 192.168.1.24
```

Using the example above your clients all mount using the same server name -

```
mount -t nfs -o vers=3 gluster:/<volume_name> <mount_dir>
```

or for CIFS -

```
net use <device> \\gluster\<sharename>
```

and those mounts are evenly spread across your cluster. That zone file fragment also sets the TTL for those records to one second so that in the unlikely event that the VIP your client is attempting to mount with is down the next retry should be to a different server.

More information

- Gluster – http://gluster.com

- CTDB - http://ctdb.samba.org

- Troubleshooting CTDB - http://ctdb.samba.org/testing.html

- Round-robin DNS - http://www.zytrax.com/books/dns/ch9/rr.html

- Setting TTL on individual resources - http://docstore.mik.ua/orelly/networking_2ndEd/dns/ch08_04.htm