Bulding a CTDB-cluster with GlusterFS

Stefan Kania

26. Mai 2020

Introduction

Why using a Cluster?

- Eliminate *single point of failure*
- *failover* if one node failed
- *Loadbalancing*
- Maintanance with out downtime

Introduction

Why using a Cluster?

- Eliminate *single point of failure*
- *failover* if one node failed
- *Loadbalancing*
- Maintanance with out downtime

Introduction

Why using a Cluster?

- Eliminate *single point of failure*
- *failover* if one node failed
- *Loadbalancing*
- Maintanance with out downtime

Introduction

Why using a Cluster?

- Eliminate *single point of failure*
- *failover* if one node failed
- *Loadbalancing*
- Maintanance with out downtime

Introduction

What do you need

- At least two nodes to build a Gluster-cluster
- Two nodes for CTDB
- Actual Samba- and GlusterFS-packages

- Today we will use Samba 4.12 and GlusterFS 7.x

Introduction

What do you need

- At least two nodes to build a Gluster-cluster
- Two nodes for CTDB
- Actual Samba- and GlusterFS-packages

- Today we will use Samba 4.12 and GlusterFS 7.x

Introduction

What do you need

- At least two nodes to build a Gluster-cluster
- Two nodes for CTDB
- Actual Samba- and GlusterFS-packages

- Today we will use Samba 4.12 and GlusterFS 7.x

Introduction

What do you need

- At least two nodes to build a Gluster-cluster
- Two nodes for CTDB
- Actual Samba- and GlusterFS-packages

- Today we will use Samba 4.12 and GlusterFS 7.x

GlusterFS

Characteristics GlusterFS

- Merge diskspace from many nodes to one volume
- Use Gluster volumes via *fusemount* over the network
- Expandable without downtime
- PosixACL Support
- Different configurations
- Self-healing
- Supporting volume snapshots

GlusterFS

Characteristics GlusterFS

- Merge diskspace from many nodes to one volume
- Use Gluster volumes via *fusemount* over the network
- Expandable without downtime
- PosixACL Support
- Different configurations
- Self-healing
- Supporting volume snapshots

GlusterFS

Characteristics GlusterFS

- Merge diskspace from many nodes to one volume
- Use Gluster volumes via *fusemount* over the network
- Expandable without downtime
- PosixACL Support
- Different configurations
- Self-healing
- Supporting volume snapshots

GlusterFS

Characteristics GlusterFS

- Merge diskspace from many nodes to one volume
- Use Gluster volumes via *fusemount* over the network
- Expandable without downtime
- PosixACL Support
- Different configurations
- Self-healing
- Supporting volume snapshots

GlusterFS

Characteristics GlusterFS

- Merge diskspace from many nodes to one volume
- Use Gluster volumes via *fusemount* over the network
- Expandable without downtime
- PosixACL Support
- Different configurations
- Self-healing
- Supporting volume snapshots

GlusterFS

Characteristics GlusterFS

- Merge diskspace from many nodes to one volume
- Use Gluster volumes via *fusemount* over the network
- Expandable without downtime
- PosixACL Support
- Different configurations
- Self-healing
- Supporting volume snapshots

GlusterFS

Characteristics GlusterFS

- Merge diskspace from many nodes to one volume
- Use Gluster volumes via *fusemount* over the network
- Expandable without downtime
- PosixACL Support
- Different configurations
- Self-healing
- Supporting volume snapshots

GlusterFS

Different volume types

- Replicated Volume
- Distributed Volume
- Striped Volume
- Replicated-Distributed Volume

- Today we will create a two node replicated volume.

GlusterFS

Different volume types

- Replicated Volume
- Distributed Volume
- Striped Volume
- Replicated-Distributed Volume

- Today we will create a two node replicated volume.

GlusterFS

Different volume types

- Replicated Volume
- Distributed Volume
- Striped Volume
- Replicated-Distributed Volume

- Today we will create a two node replicated volume.

GlusterFS

Different volume types

- Replicated Volume
- Distributed Volume
- Striped Volume
- Replicated-Distributed Volume

- Today we will create a two node replicated volume.

GlusterFS

Different volume types

- Replicated Volume
- Distributed Volume
- Striped Volume
- Replicated-Distributed Volume

- Today we will create a two node replicated volume.

GlusterFS

Setting up GlusterFS

- Installing the package glusterfs-server (already done)
- Configure trusted pool with peer probe
- Creating the brick with *LVM2* (already done)
- Formatting the brick with *xfs* (already done)
- Mount the brick to /gluster (already done)
- Creat the volume with volume create
- Start the Volumes with volume start
- Mount the volume via systemd

GlusterFS

Setting up GlusterFS

- Installing the package glusterfs-server (already done)
- Configure trusted pool with peer probe
- Creating the brick with *LVM2* (already done)
- Formatting the brick with *xfs* (already done)
- Mount the brick to /gluster (already done)
- Creat the volume with volume create
- Start the Volumes with volume start
- Mount the volume via systemd

GlusterFS

Setting up GlusterFS

- Installing the package glusterfs-server (already done)
- Configure trusted pool with peer probe
- Creating the brick with *LVM2* (already done)
- Formatting the brick with *xfs* (already done)
- Mount the brick to /gluster (already done)
- Creat the volume with volume create
- Start the Volumes with volume start
- Mount the volume via systemd

GlusterFS

Setting up GlusterFS

- Installing the package glusterfs-server (already done)
- Configure trusted pool with peer probe
- Creating the brick with *LVM2* (already done)
- Formatting the brick with *xfs* (already done)
- Mount the brick to /gluster (already done)
- Creat the volume with volume create
- Start the Volumes with volume start
- Mount the volume via systemd

GlusterFS

Setting up GlusterFS

- Installing the package glusterfs-server (already done)
- Configure trusted pool with peer probe
- Creating the brick with *LVM2* (already done)
- Formatting the brick with *xfs* (already done)
- Mount the brick to /gluster (already done)
- Creat the volume with volume create
- Start the Volumes with volume start
- Mount the volume via systemd

GlusterFS

Setting up GlusterFS

- Installing the package glusterfs-server (already done)
- Configure trusted pool with peer probe
- Creating the brick with *LVM2* (already done)
- Formatting the brick with *xfs* (already done)
- Mount the brick to /gluster (already done)
- Creat the volume with volume create
- Start the Volumes with volume start
- Mount the volume via systemd

GlusterFS

Setting up GlusterFS

- Installing the package glusterfs-server (already done)
- Configure trusted pool with peer probe
- Creating the brick with *LVM2* (already done)
- Formatting the brick with *xfs* (already done)
- Mount the brick to /gluster (already done)
- Creat the volume with volume create
- Start the Volumes with volume start
- Mount the volume via systemd

GlusterFS

Setting up GlusterFS

- Installing the package glusterfs-server (already done)
- Configure trusted pool with peer probe
- Creating the brick with *LVM2* (already done)
- Formatting the brick with *xfs* (already done)
- Mount the brick to /gluster (already done)
- Creat the volume with volume create
- Start the Volumes with volume start
- Mount the volume via systemd

CTDB

Characteristics of CTDB

- tdb-Databases can keep consistent over all nodes
- Automatic repair of tdb-databases after crash
- CTDB brings HA-functionality, failover and IP-takeover
- Loadbalancing via Round Robin DNS

CTDB

Characteristics of CTDB

- tdb-Databases can keep consistent over all nodes
- Automatic repair of tdb-databases after crash
- CTDB brings HA-functionality, failover and IP-takeover
- Loadbalancing via Round Robin DNS

CTDB

Characteristics of CTDB

- tdb-Databases can keep consistent over all nodes
- Automatic repair of tdb-databases after crash
- CTDB brings HA-functionality, failover and IP-takeover
- Loadbalancing via Round Robin DNS

CTDB

Characteristics of CTDB

- tdb-Databases can keep consistent over all nodes
- Automatic repair of tdb-databases after crash
- CTDB brings HA-functionality, failover and IP-takeover
- Loadbalancing via Round Robin DNS

CTDB

Configuring of CTDB

- Installing Samba-packages (already done)
- Creat DNS-entries for the cluster
- Configuring *recovery lock* in /etc/ctdb/ctdb.conf
  - recovery lock = /glusterfs/ctdb.lock
- Creat the files nodes and public_addresses on all nodes
- Start the ctdb-service on all nodes
- Test the service with `ctdb status`

CTDB

Configuring of CTDB

- Installing Samba-packages (already done)
- Creat DNS-entries for the cluster
- Configuring *recovery lock* in /etc/ctdb/ctdb.conf
  - recovery lock = /glusterfs/ctdb.lock
- Creat the files nodes and public_addresses on all nodes
- Start the ctdb-service on all nodes
- Test the service with ctdb status

CTDB

Configuring of CTDB

- Installing Samba-packages (already done)
- Creat DNS-entries for the cluster
- Configuring *recovery lock* in /etc/ctdb/ctdb.conf
  - recovery lock = /glusterfs/ctdb.lock
- Creat the files nodes and public_addresses on all nodes
- Start the ctdb-service on all nodes
- Test the service with `ctdb status`

CTDB

Configuring of CTDB

- Installing Samba-packages (already done)
- Creat DNS-entries for the cluster
- Configuring *recovery lock* in /etc/ctdb/ctdb.conf
  - recovery lock = /glusterfs/ctdb.lock
- Creat the files nodes and public_addresses on all nodes
- Start the ctdb-service on all nodes
- Test the service with ctdb status

CTDB

Configuring of CTDB

- Installing Samba-packages (already done)
- Creat DNS-entries for the cluster
- Configuring *recovery lock* in /etc/ctdb/ctdb.conf
  - recovery lock = /glusterfs/ctdb.lock
- Creat the files nodes and public_addresses on all nodes
- Start the ctdb-service on all nodes
- Test the service with ctdb status

CTDB

Configuring of CTDB

- Installing Samba-packages (already done)
- Creat DNS-entries for the cluster
- Configuring *recovery lock* in /etc/ctdb/ctdb.conf
  - recovery lock = /glusterfs/ctdb.lock
- Creat the files nodes and public_addresses on all nodes
- Start the ctdb-service on all nodes
- Test the service with ctdb status

CTDB

Configuring of CTDB

- Installing Samba-packages (already done)
- Creat DNS-entries for the cluster
- Configuring *recovery lock* in /etc/ctdb/ctdb.conf
  - recovery lock = /glusterfs/ctdb.lock
- Creat the files nodes and public_addresses on all nodes
- Start the ctdb-service on all nodes
- Test the service with ctdb status

CTDB

Configuring of CTDB

- Installing Samba-packages (already done)
- Creat DNS-entries for the cluster
- Configuring *recovery lock* in /etc/ctdb/ctdb.conf
  - recovery lock = /glusterfs/ctdb.lock
- Creat the files nodes and public_addresses on all nodes
- Start the ctdb-service on all nodes
- Test the service with ctdb status

CTDB

Configuring of CTDB

- Installing Samba-packages (already done)
- Creat DNS-entries for the cluster
- Configuring *recovery lock* in /etc/ctdb/ctdb.conf
  - recovery lock = /glusterfs/ctdb.lock
- Creat the files nodes and public_addresses on all nodes
- Start the ctdb-service on all nodes
- Test the service with `ctdb status`

CTDB

Content of file »nodes«

192.168.57.42
192.168.57.43

Inhalt der Datei »public_addresses«

192.168.56.101/24 enp0s8
192.168.56.102/24 enp0s8

CTDB

Output from »ctdb status«

```
root@samba42-fs1:# ctdb status
Number of nodes:2
pnn:0 192.168.56.101 OK (THIS NODE)
pnn:1 192.168.56.102 OK
Generation:446598079
Size:2
hash:0 lmaster:0
hash:1 lmaster:1
Recovery mode:NORMAL (0)
Recovery master:0
```

CTDB

Managing Samba-services via CTDB

- Create a /etc/samba/smb.conf with the following content

```
[global]
  clustering = yes
  include = registry
```

- Import the configuration into the *Registry*
- Remmove smbd, nmbd and winbind from systemd
- Enable Samba ctdb event script enable legacy 50.samba
- Enable winbind ctdb event script enable legacy 49.winbind
- Join the domain
- Restart CTDB on all nodes

CTDB

Managing Samba-services via CTDB

- Create a /etc/samba/smb.conf with the following content

```
[global]
  clustering = yes
  include = registry
```

- Import the configuration into the *Registry*
- Remmove smbd, nmbd and winbind from systemd
- Enable Samba ctdb event script enable legacy 50.samba
- Enable winbind ctdb event script enable legacy 49.winbind
- Join the domain
- Restart CTDB on all nodes

CTDB

Managing Samba-services via CTDB

- Create a /etc/samba/smb.conf with the following content

```
[global]
  clustering = yes
  include = registry
```

- Import the configuration into the *Registry*
- Remmove smbd, nmbd and winbind from systemd
- Enable Samba ctdb event script enable legacy 50.samba
- Enable winbind ctdb event script enable legacy 49.winbind
- Join the domain
- Restart CTDB on all nodes

CTDB

Managing Samba-services via CTDB

- Create a /etc/samba/smb.conf with the following content

```
[global]
  clustering = yes
  include = registry
```

- Import the configuration into the *Registry*
- Remmove smbd, nmbd and winbind from systemd
- Enable Samba ctdb event script enable legacy 50.samba
- Enable winbind ctdb event script enable legacy 49.winbind
- Join the domain
- Restart CTDB on all nodes

CTDB

Managing Samba-services via CTDB

- Create a /etc/samba/smb.conf with the following content

```
[global]
  clustering = yes
  include = registry
```

- Import the configuration into the *Registry*
- Remmove smbd, nmbd and winbind from systemd
- Enable Samba ctdb event script enable legacy 50.samba
- Enable winbind ctdb event script enable legacy 49.winbind
- Join the domain
- Restart CTDB on all nodes

CTDB

Managing Samba-services via CTDB

- Create a /etc/samba/smb.conf with the following content

```
[global]
  clustering = yes
  include = registry
```

- Import the configuration into the *Registry*
- Remmove smbd, nmbd and winbind from systemd
- Enable Samba ctdb event script enable legacy 50.samba
- Enable winbind ctdb event script enable legacy 49.winbind
- Join the domain
- Restart CTDB on all nodes

CTDB

Managing Samba-services via CTDB

- Create a /etc/samba/smb.conf with the following content

```
[global]
  clustering = yes
  include = registry
```

- Import the configuration into the *Registry*
- Remmove smbd, nmbd and winbind from systemd
- Enable Samba ctdb event script enable legacy 50.samba
- Enable winbind ctdb event script enable legacy 49.winbind
- Join the domain
- Restart CTDB on all nodes

CTDB

Managing Samba-services via CTDB

- Create a /etc/samba/smb.conf with the following content

```
[global]
  clustering = yes
  include = registry
```

- Import the configuration into the *Registry*
- Remmove smbd, nmbd and winbind from systemd
- Enable Samba ctdb event script enable legacy 50.samba
- Enable winbind ctdb event script enable legacy 49.winbind
- Join the domain
- Restart CTDB on all nodes

CTDB

Managing Samba-services via CTDB

- Create a /etc/samba/smb.conf with the following content

```
[global]
  clustering = yes
  include = registry
```

- Import the configuration into the *Registry*
- Remmove smbd, nmbd and winbind from systemd
- Enable Samba ctdb event script enable legacy 50.samba
- Enable winbind ctdb event script enable legacy 49.winbind
- Join the domain
- Restart CTDB on all nodes

CTDB

Test the CTDB-cluster

- ctdb event status legacy monitor (see running services)
- ctdb event script list legacy (see all services)
- ctdb ping -n all
- ctdb ip
- ctdb ipinfo 192.168.56.101 (dynamic IP)

CTDB

Test the CTDB-cluster

- ctdb event status legacy monitor (see running services)
- ctdb event script list legacy (see all services)
- ctdb ping -n all
- ctdb ip
- ctdb ipinfo 192.168.56.101 (dynamic IP)

CTDB

Test the CTDB-cluster

- ctdb event status legacy monitor (see running services)
- ctdb event script list legacy (see all services)
- ctdb ping -n all
- ctdb ip
- ctdb ipinfo 192.168.56.101 (dynamic IP)

CTDB

Test the CTDB-cluster

- ctdb event status legacy monitor (see running services)
- ctdb event script list legacy (see all services)
- ctdb ping -n all
- ctdb ip
- ctdb ipinfo 192.168.56.101 (dynamic IP)

CTDB

Test the CTDB-cluster

- ctdb event status legacy monitor (see running services)
- ctdb event script list legacy (see all services)
- ctdb ping -n all
- ctdb ip
- ctdb ipinfo 192.168.56.101 (dynamic IP)

onnode

Manage the cluster with onnode

- onnode all ctdb status
- onnode all service ctdb restart
- onnode all cp /glusterfs/nodes /etc/ctdb

onnode

Manage the cluster with onnode

- onnode all ctdb status
- onnode all service ctdb restart
- onnode all cp /glusterfs/nodes /etc/ctdb

onnode

Manage the cluster with onnode

- onnode all ctdb status
- onnode all service ctdb restart
- onnode all cp /glusterfs/nodes /etc/ctdb

Shares

Three ways to create shares

- The classic way via a mounted Gluster-volume
- Using the VFS-module glusterfs
- Using the VFS-module glusterfs_fuse

Shares

Three ways to create shares

- The classic way via a mounted Gluster-volume
- Using the VFS-module glusterfs
- Using the VFS-module glusterfs_fuse

Shares

Three ways to create shares

- The classic way via a mounted Gluster-volume
- Using the VFS-module glusterfs
- Using the VFS-module glusterfs_fuse

Let's go

and let us start the show