

Samba

Witness Protection Programming

Samuel Cabrero
scabrero@suse.com
David Disseldorp
ddiss@samba.org

Agenda

- Clustered Samba
- Witness Protocol
- Demo
- Outlook



Clustered Samba

The background features abstract geometric shapes. A large teal shape occupies the left side, while a green shape is on the right. A white diagonal line separates them, and a white horizontal line is visible at the top right.

Samba

- File and print server
 - SMB / CIFS, SMB2 and SMB3+ dialects
- Authentication
 - NTLMv2 and Kerberos
- Identity mapping
 - Windows *SIDs* to *uids* and *gids*
 - Active Directory domain member or domain controller



Samba

- *smbd*
 - Main server daemon
 - Spawns separate processes for various RPC services
 - Endpoint mapper, spoolss (printing), etc.
 - Forked for each client connection
 - Pluggable back-end
 - *vfs_btrfs*, *vfs_ceph*, *vfs_fruit*, etc.
- *winbindd*
 - Authentication and ID mapping daemon
 - Communicates with AD DC when joined to a domain



Samba State

- SMB protocol requires server state tracking
 - Client connections
 - Open files, locks and leases
- Samba relies on Trivial Database (TDB)
 - Key-value store
 - Supports multiple writers, record locking, transactions, etc.
 - Single node only



Clustering with CTDB

- Clustered Trivial Database (CTDB)
- Share state across multiple Samba nodes
 - Volatile and persistent databases
 - Reliable messaging
- Active / Active
- HA features
 - Monitoring and failover



Clustering with CTDB

- Record location master and data master
 - Location determined by hash of key and active node map
- Elected recovery master monitors state of cluster
- Performs database recovery if necessary
 - Cluster-wide mutex used to prevent split brain
- “Tickle” clients on IP failover



Clustering in Windows and Samba

- Windows
 - File Server role (active / passive)
 - Scaleout File Server role (active / active)
- Samba + CTDB
 - All nodes are active at the same time
 - Clients access the cluster by the public IP addresses pool, distributed dynamically between nodes (floating IPs)



Witness Protocol

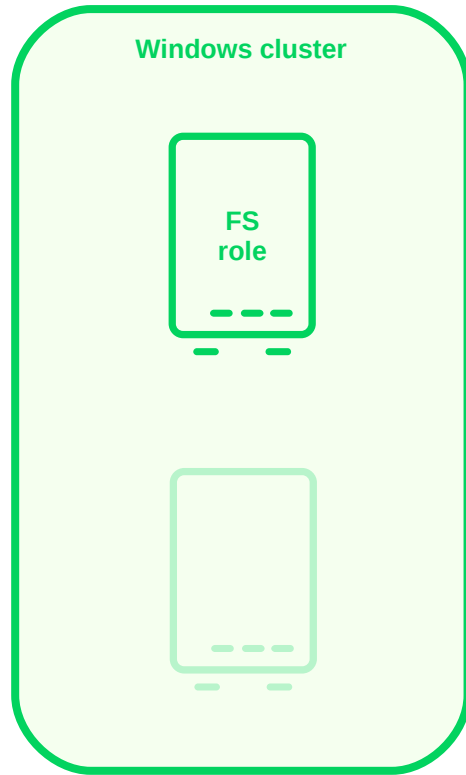
The background features a large teal shape on the left and a green shape on the right, separated by a white diagonal line. The teal shape is a large, irregular polygon with a pointed top and a pointed bottom. The green shape is a large, irregular polygon with a pointed top and a pointed bottom, mirroring the teal shape's orientation. The white diagonal line runs from the top right towards the bottom left, creating a clear division between the two colored areas.

Witness protocol

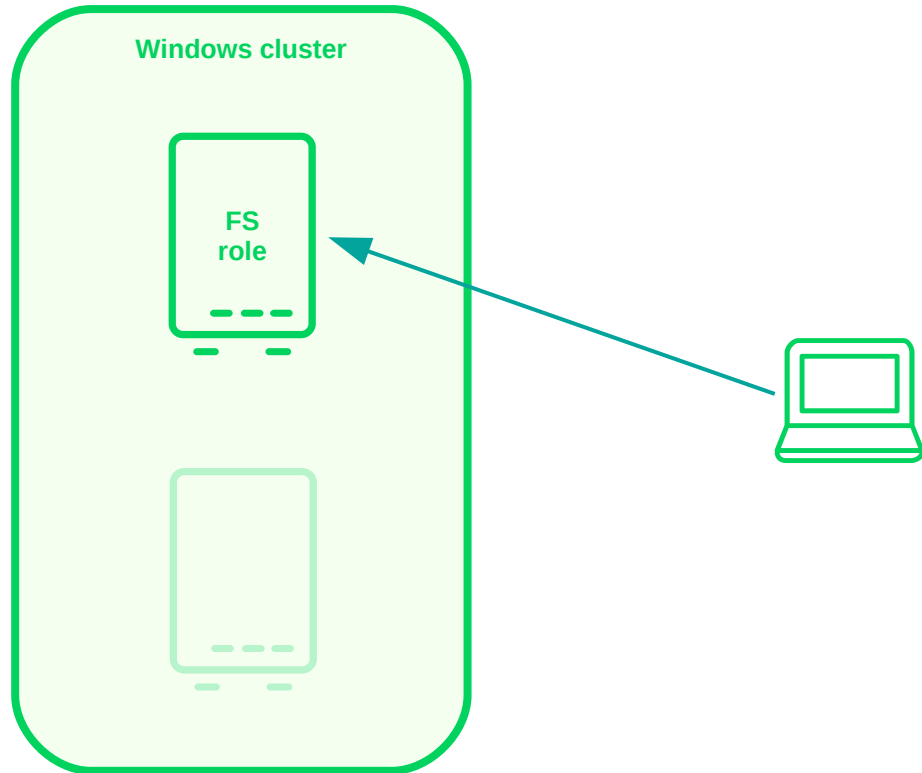
- Advertise cluster state changes to clients
- Transparent client failover
- Load balancing
- Allows continuous availability of SMB shares in clustered environments
- Runs as a DCE/RPC service



SMB1 and SMB2 client failover

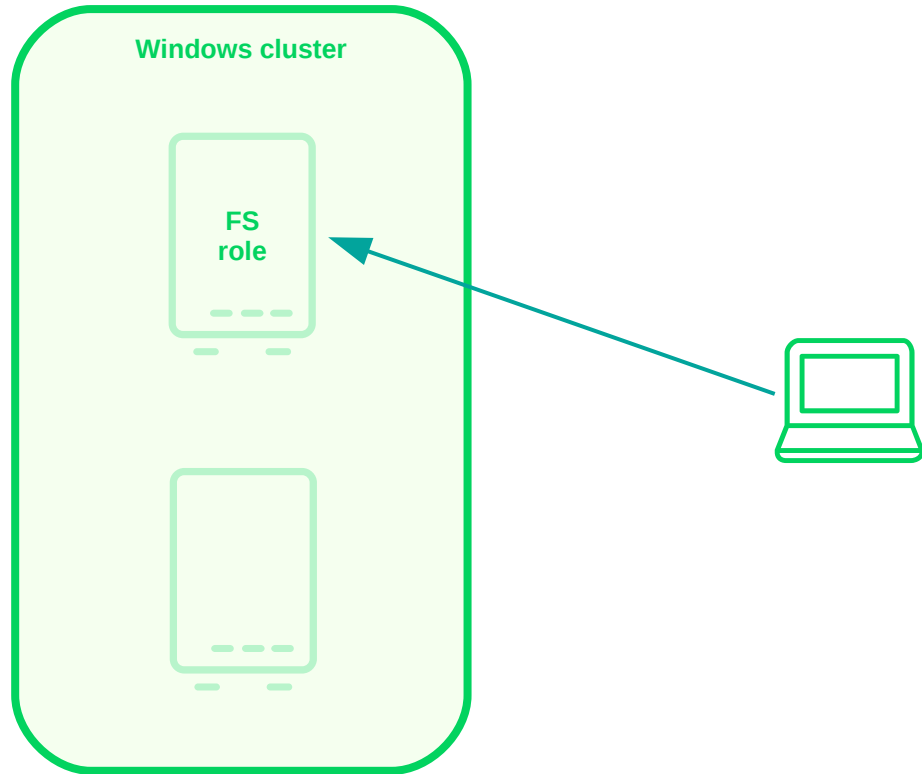


SMB1 and SMB2 client failover



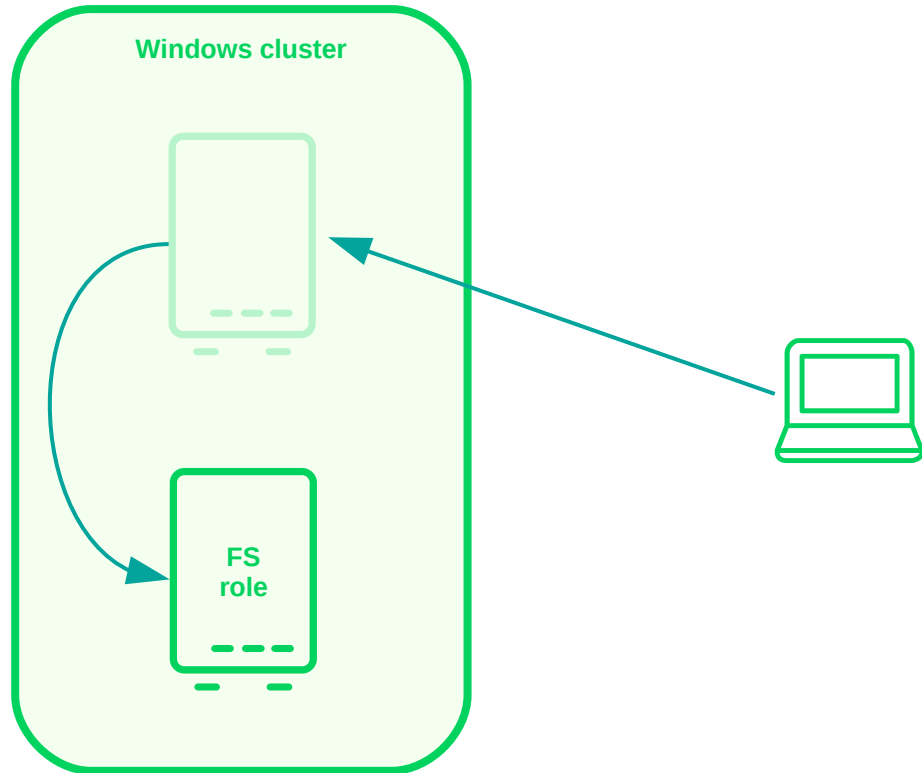
- Client opens SMB connection

SMB1 and SMB2 client failover



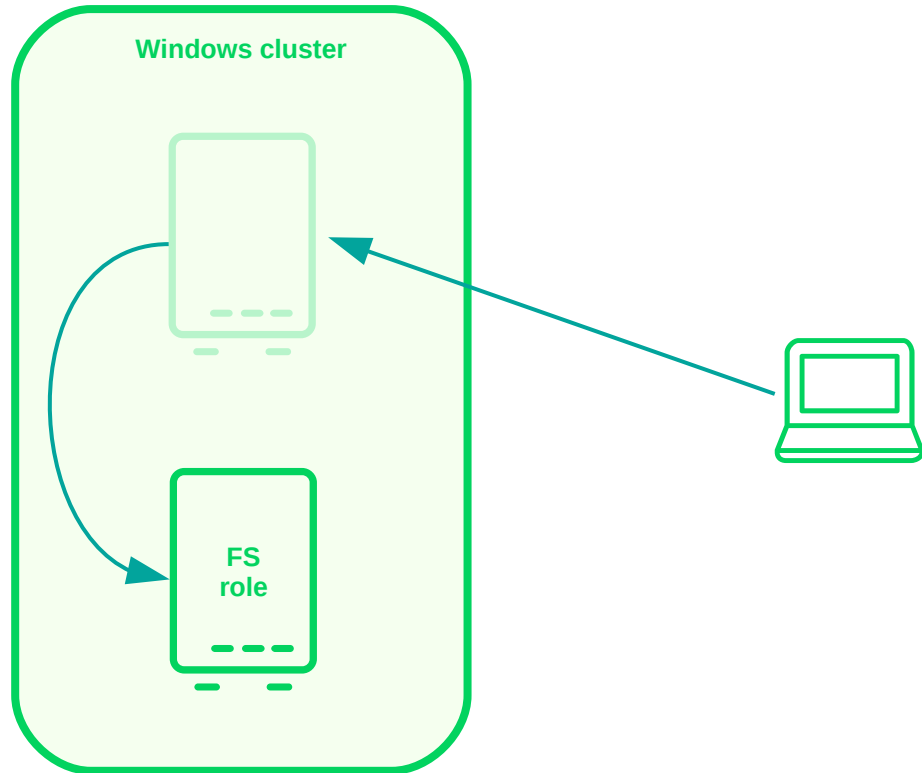
- Client opens SMB connection to node 1
- Node 1 goes down

SMB1 and SMB2 client failover



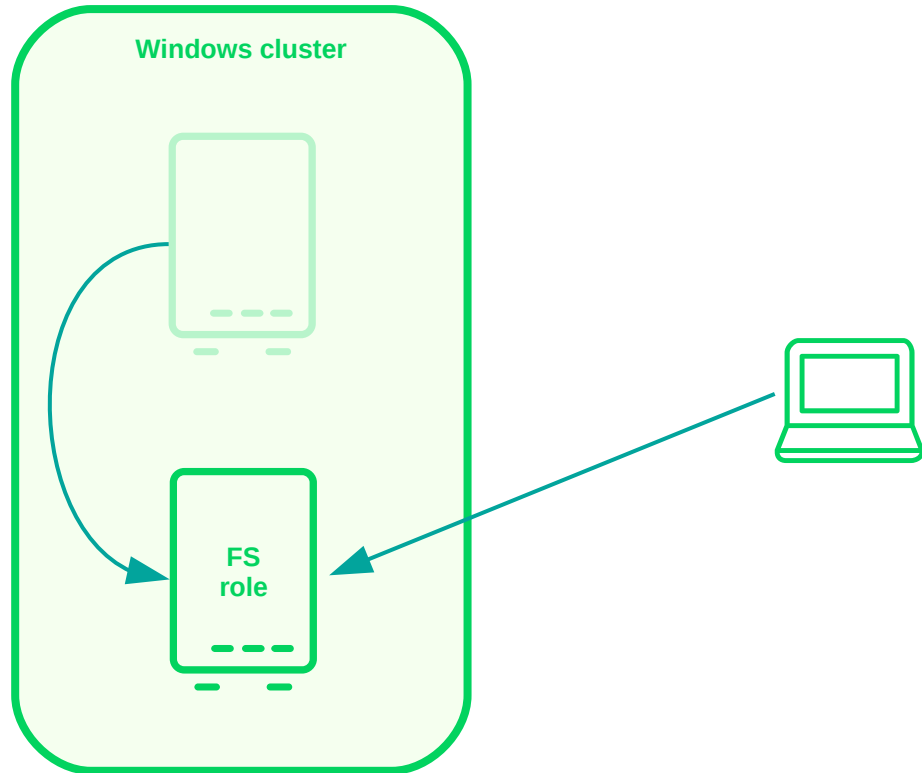
- Client opens SMB connection to node 1
- Node 1 goes down
- Node 2 takes the role

SMB1 and SMB2 client failover



- Client opens SMB connection to node 1
- Node 1 goes down
- Node 2 takes the role
- Client waits TCP timeout...

SMB1 and SMB2 client failover



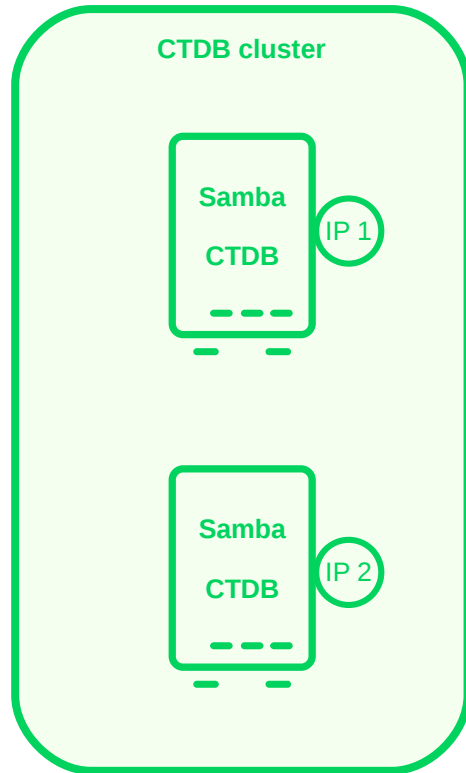
- Client opens SMB connection to node 1
- Node 1 goes down
- Node 2 takes the role
- Client waits TCP timeout...
- Client reconnects

SMB1 and SMB2 client failover with CTDB

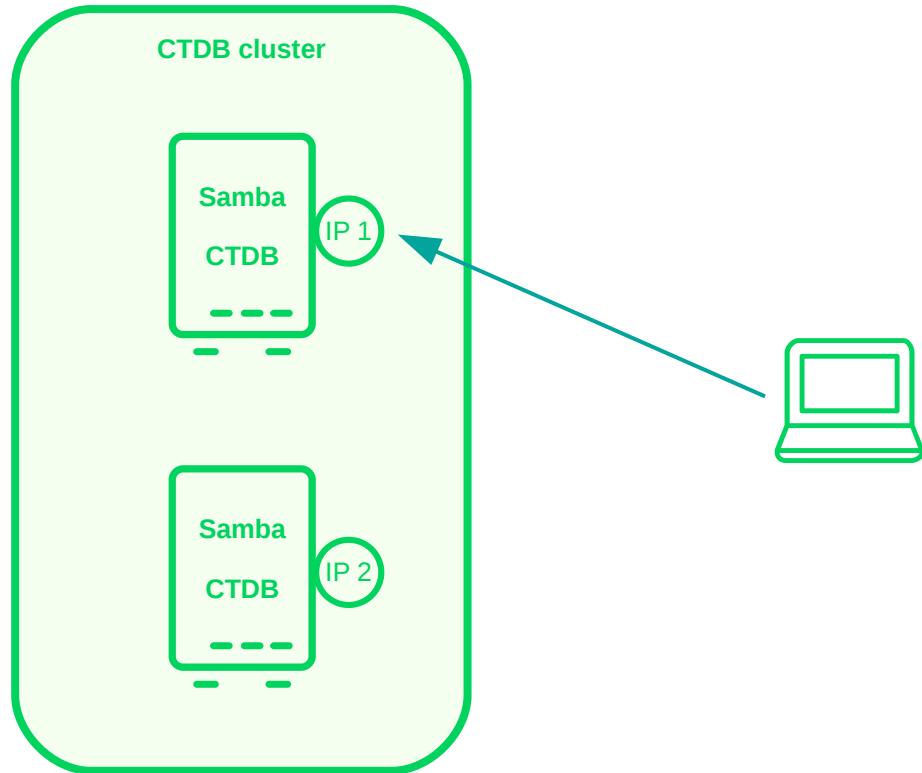
- IP takeover
- Uses “Tickle ACKs” and “gratuitous ARP” to speedup recovery



SMB1 and SMB2 client failover with CTDB

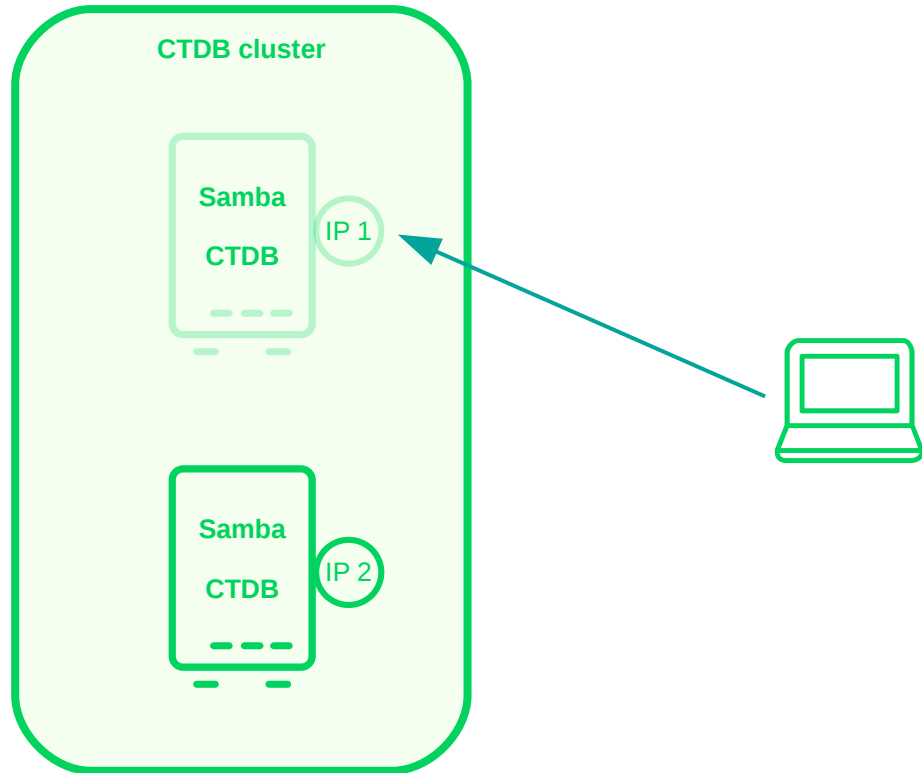


SMB1 and SMB2 client failover with CTDB



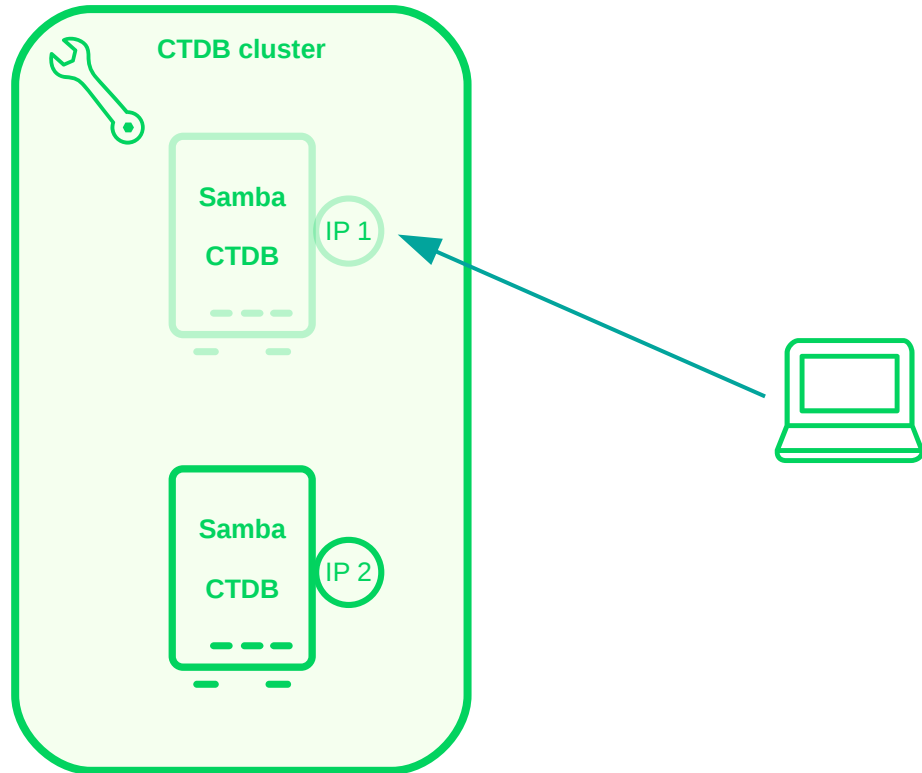
- Client opens SMB connection

SMB1 and SMB2 client failover with CTDB



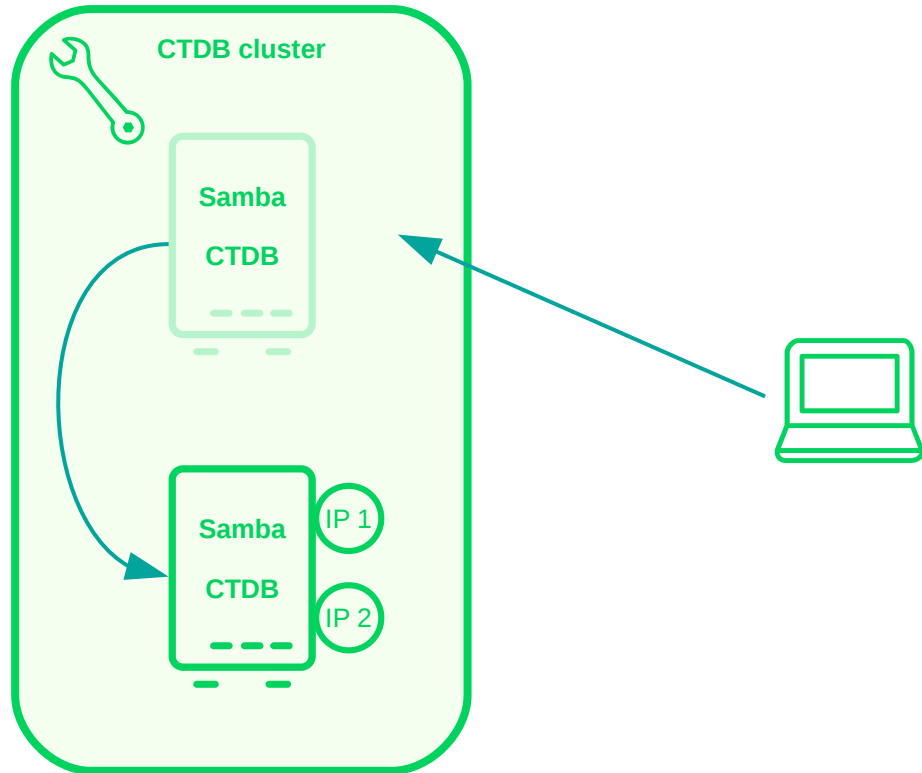
- Client opens SMB connection
- Node goes down

SMB1 and SMB2 client failover with CTDB



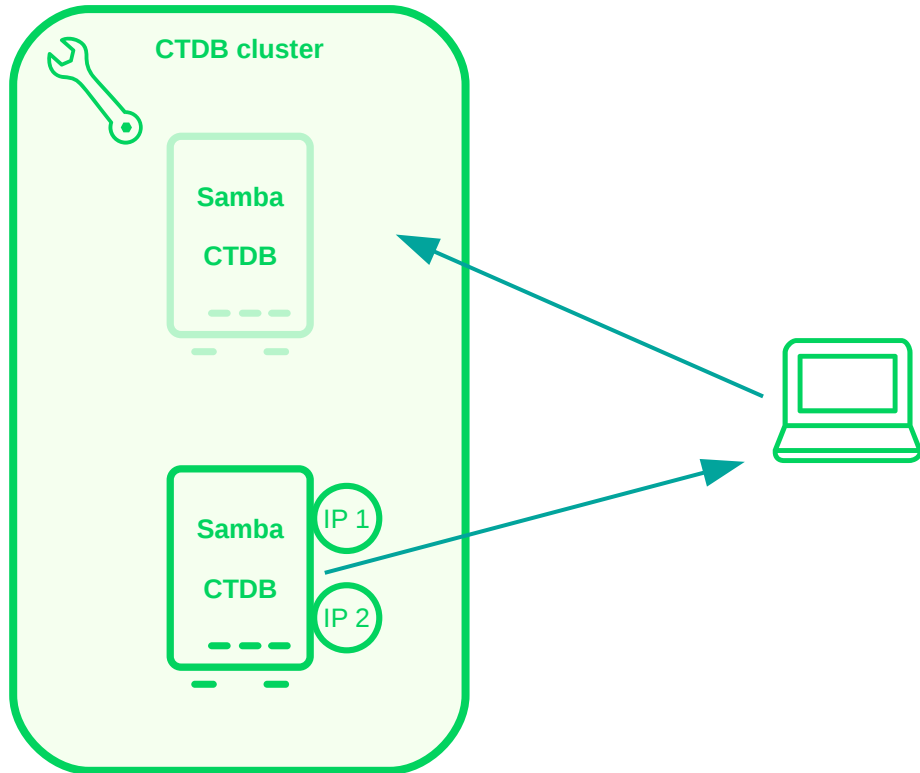
- Client opens SMB connection
- Node goes down
- CTDB enters in recovery and runs IP takeover

SMB1 and SMB2 client failover with CTDB



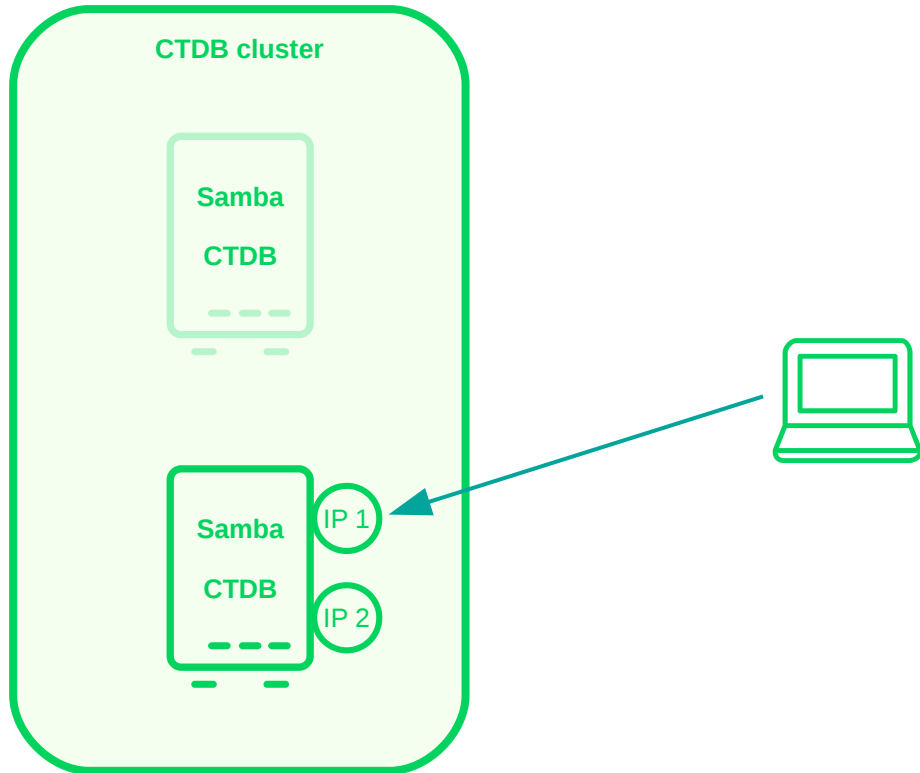
- Client opens SMB connection
- Node goes down
- CTDB enters in recovery and runs IP takeover
- CTDB “takes” the unavailable IPs

SMB1 and SMB2 client failover with CTDB



- Client opens SMB connection
- Node goes down
- CTDB enters in recovery and runs IP takeover
- CTDB “takes” the unavailable Ips
- CTDB sends to all clients that were connected to node 1 a “gratuitous ARP” and a “tickle ACK”

SMB1 and SMB2 client failover with CTDB



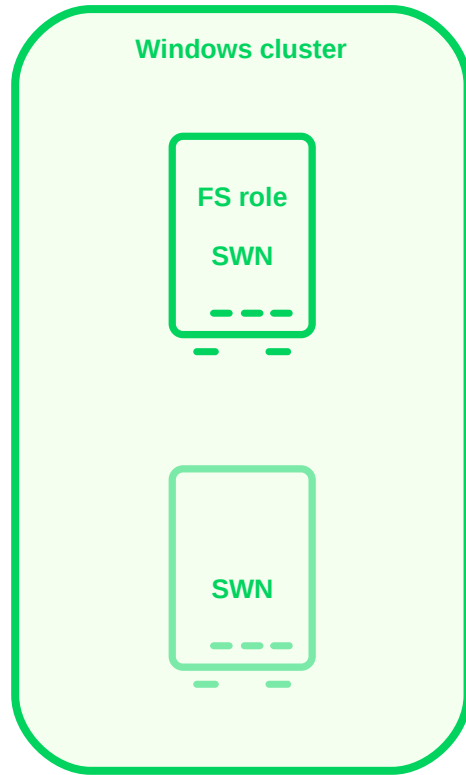
- Client opens SMB connection
- Node goes down
- CTDB enters in recovery and runs IP takeover
- CTDB “takes” the unavailable Ips
- CTDB sends to all clients that were connected to node 1 a “gratuitous ARP” and a “tickle ACK”
- The tickle ACK resets the connection

SMB3 client failover

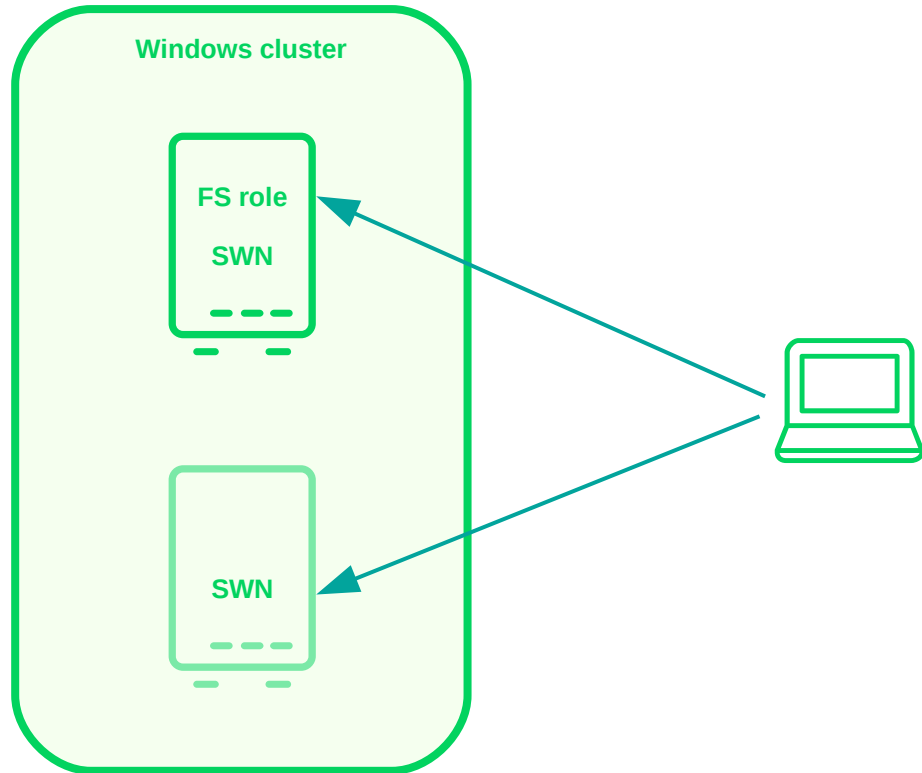
- Transparent thanks to several new features
 - Persistent handles
 - Part of SMB3 protocol
 - Server maintains file handle state and persist it
 - If client/server crashes the client can reestablish the file handle state
 - Work in progress
 - Witness
 - New protocol independent from SMB



SMB3 client failover

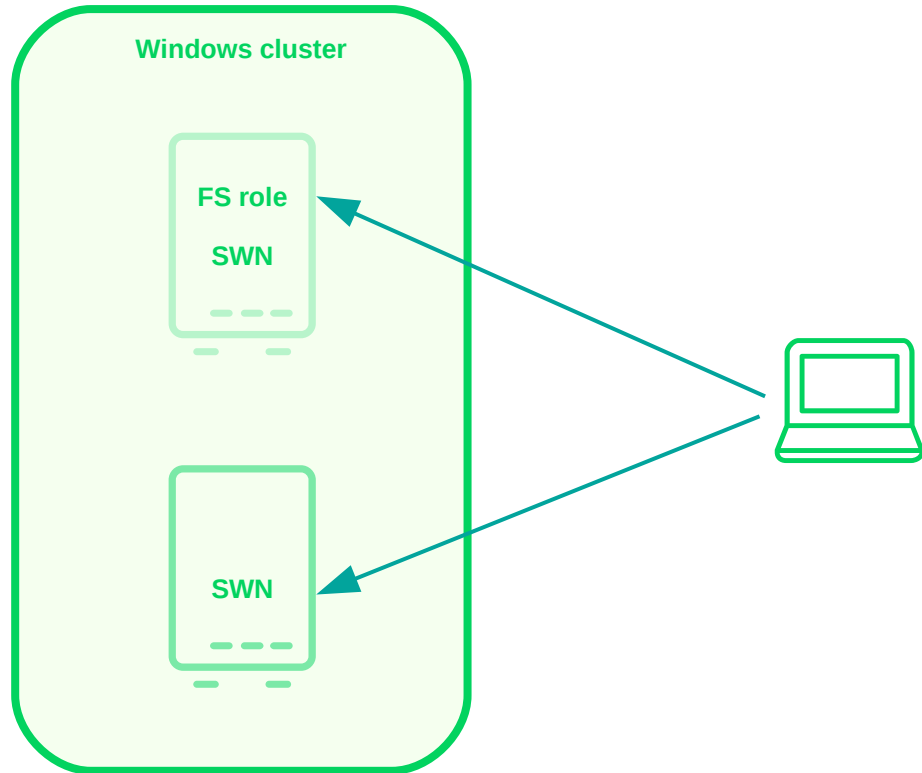


SMB3 client failover



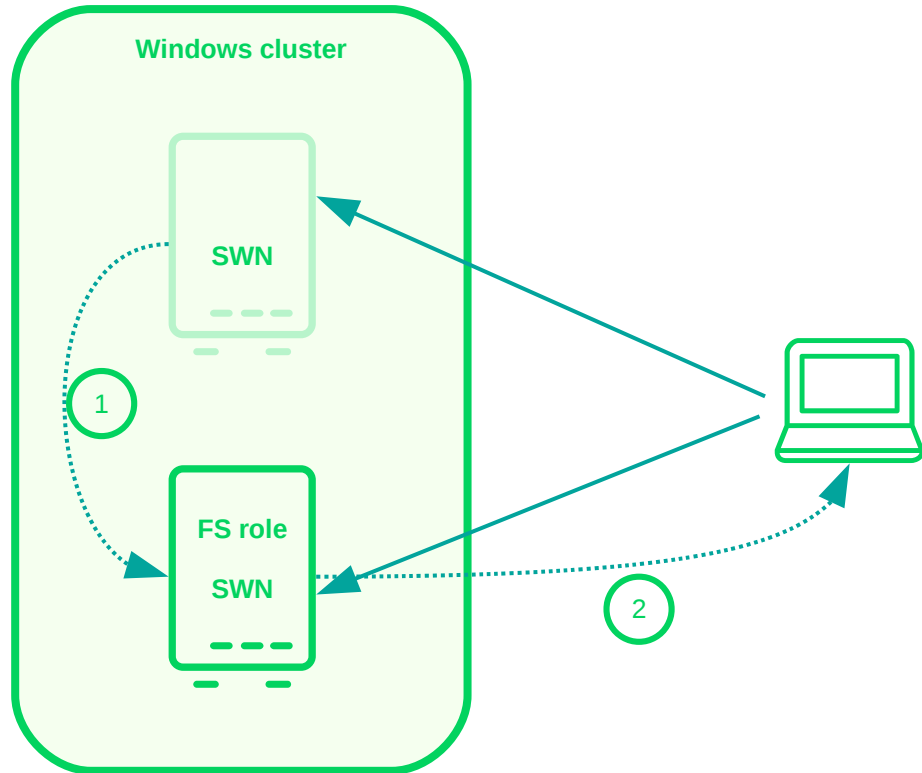
- Client opens SMB and SWN connections, always on different nodes

SMB3 client failover



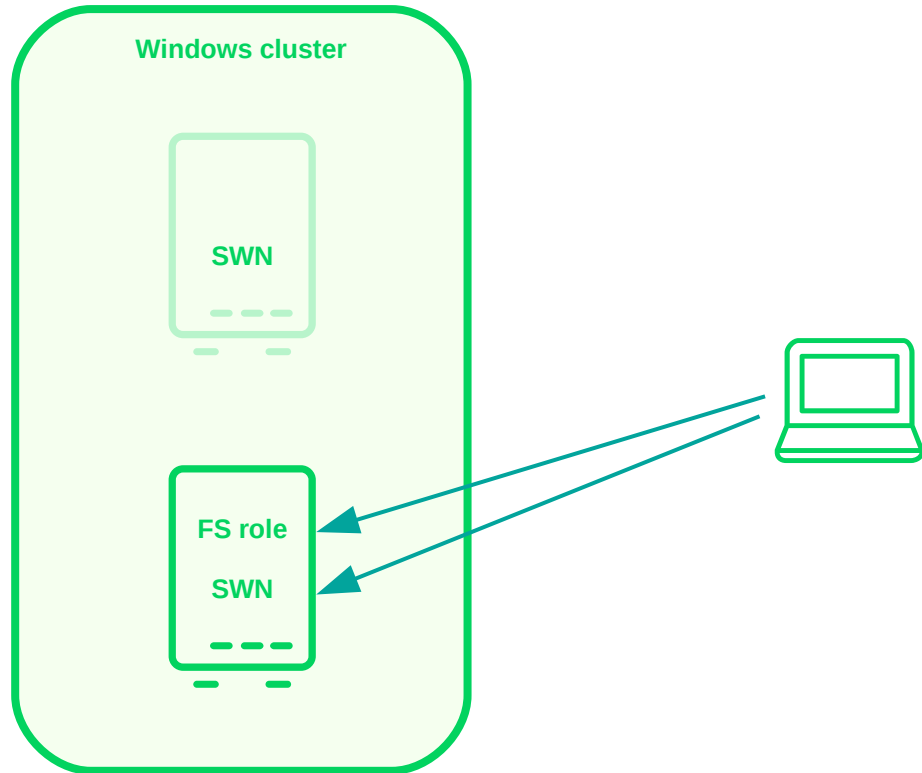
- Client opens SMB and SWN connections, always on different nodes
- Node goes down

SMB3 client failover



- Client opens SMB and SWN connections, always on different nodes
- Node goes down
- Node 2 takes the role and notify the client role is now in the node 2

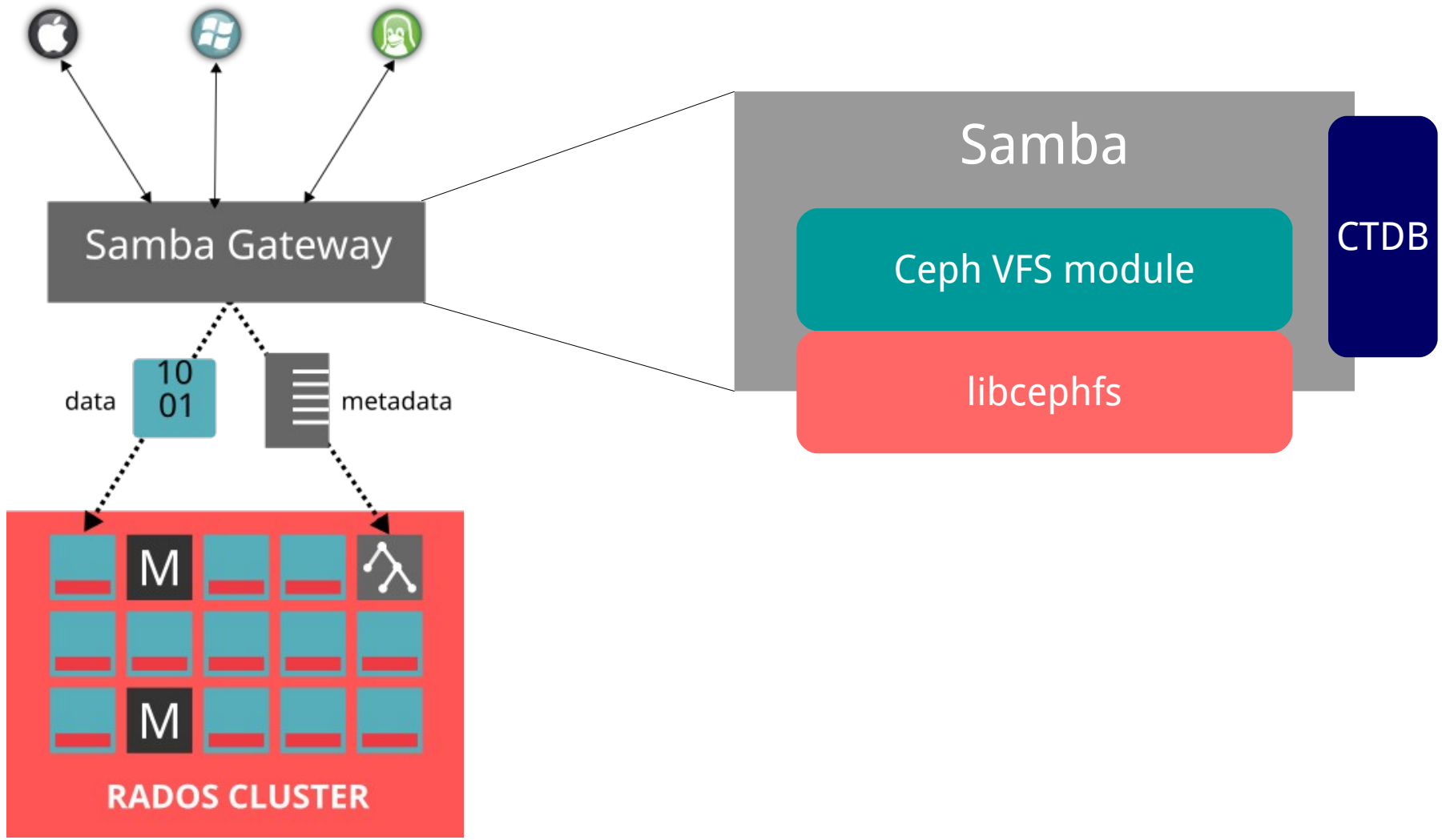
SMB3 client failover



- Client opens SMB and SWN connections, always on different nodes
- Node goes down
- Node 2 takes the role and notify the client role is now in the node 2
- Client reacts to notification and reconnects to node 2



Demo



Demo: Samba + CTDB + CephFS

- CephFS module for Samba: *vfs_ceph*
- Ceph RADOS clustered mutex helper for CTDB
- Asynchronous DCE/RPC server
- Witness server
- Persistent Handles



Future Outlook

The background features abstract geometric shapes in two shades of green. A large teal shape occupies the left and top portions, while a bright green shape is on the right. A white diagonal line separates the two green areas, and a white horizontal line is visible at the top right.

Work in Progress – Witness

- Upstreaming
- Protocol requires asynchronous DCE/RPC server
 - Partial rewrite samba3 DCE/RPC server
 - Merge samba4 and samba3 implementations
- Automatic cluster load balancing



Samba: Future

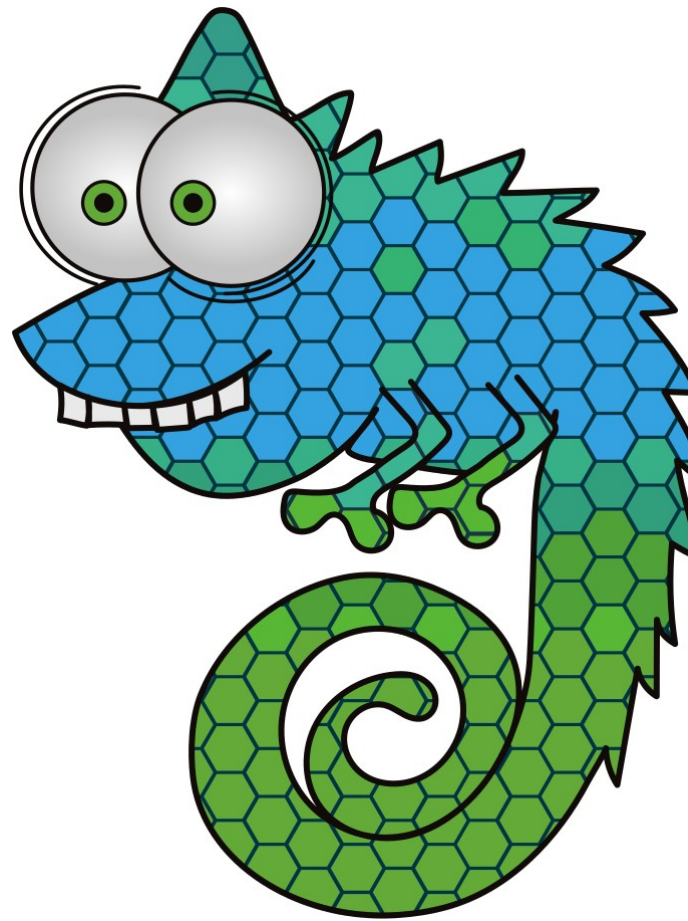
- Replace or modify CTDB
- Ceph omap backed key-value store for Samba
 - Samba database API demanding
 - Multiple processes and writers
 - Record locking and transactions
 - RADOS classes



References

- Samba: <https://samba.org/>
- CTDB: <https://ctdb.samba.org/>
- SMB 3.1.1 encryption: [https://technet.microsoft.com/en-us/library/dn551363\(v=ws.11\).aspx](https://technet.microsoft.com/en-us/library/dn551363(v=ws.11).aspx)
- Witness Protocol:
http://www.sambaxp.org/archive_data/SambaXP2015-SLIDES/wed/track1/sambaxp2015-wed-track1-Guenther_Deschner-ImplementingTheWitnessProtocolInSamba.pdf





Join Us at www.opensuse.org



License

This slide deck is licensed under the Creative Commons Attribution-ShareAlike 4.0 International license. It can be shared and adapted for any purpose (even commercially) as long as Attribution is given and any derivative work is distributed under the same license.

Details can be found at <https://creativecommons.org/licenses/by-sa/4.0/>

General Disclaimer

This document is not to be construed as a promise by any participating organisation to develop, deliver, or market a product. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. openSUSE makes no representations or warranties with respect to the contents of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. The development, release, and timing of features or functionality described for openSUSE products remains at the sole discretion of openSUSE. Further, openSUSE reserves the right to revise this document and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes. All openSUSE marks referenced in this presentation are trademarks or registered trademarks of SUSE LLC, in the United States and other countries. All third-party trademarks are the property of their respective owners.

Credits

Template

Richard Brown
rbrown@opensuse.org

Design & Inspiration

openSUSE Design Team
<http://opensuse.github.io/branding-guidelines/>