

SMB Witness Service

In Samba CTDB Clusters

Stefan Metzmacher <metze@samba.org>

Samba Team / SerNet

2024-09-18

<https://samba.org/~metze/presentations/2024/SDC/>

- ▶ What is the Service Witness Protocol [MS-SWN]
- ▶ Examples how it works
- ▶ rpcd_witness design
- ▶ Some strange things a Windows client is doing.
- ▶ How to configure rpcd_witness
- ▶ net witness commands
- ▶ Questions? Feedback!

What is the Service Witness Protocol [MS-SWN]

- ▶ The Service Witness Protocol [MS-SWN]:
 - ▶ Provides a way to notify SMB3 clients about cluster failures
 - ▶ Either network interface or node failures
 - ▶ Or planned downtimes or loadbalancing by administrators
- ▶ The protocol itself is independent of SMB3:
 - ▶ It is based on DCERPC over TCP (`ncacn_ip_tcp`)
 - ▶ It uses kerberos or NTLMSSP integrity protection

What is the Service Witness Protocol [MS-SWN]

- ▶ The Service Witness Protocol [MS-SWN]:
 - ▶ Provides a way to notify SMB3 clients about cluster failures
 - ▶ Either network interface or node failures
 - ▶ Or planned downtimes or loadbalancing by administrators
- ▶ The protocol itself is independent of SMB3:
 - ▶ It is based on DCERPC over TCP (ncacn_ip_tcp)
 - ▶ It uses kerberos or NTLMSSP integrity protection

Basic flow of a client connecting with witness

12:27:47,488023	172.31.9.118	172.31.99.168	SMB	Negotiate Protocol Request		
12:27:47,514557	172.31.99.168	172.31.9.118	SMB2	Negotiate Protocol Response	Client:	172.31.9.118
12:27:47,514719	172.31.9.118	172.31.99.168	SMB2	Negotiate Protocol Request	Node0:	172.31.99.166
12:27:47,515661	172.31.99.168	172.31.9.118	SMB2	Negotiate Protocol Response	Node1:	172.31.99.167
12:27:47,519042	172.31.9.118	172.31.99.168	SMB2	Session Setup Request	Node2:	172.31.99.168
12:27:47,783808	172.31.99.168	172.31.9.118	SMB2	Session Setup Response		
12:27:47,784356	172.31.9.118	172.31.99.168	SMB2	Tree Connect Request Tree: \\ubcluster.w2022-17.base\IPC\$		
12:27:47,786034	172.31.99.168	172.31.9.118	SMB2	Tree Connect Response		
12:27:51,604462	172.31.9.118	172.31.99.168	SMB2	Tree Connect Request Tree: \\ubcluster.w2022-17.base\shm		
12:27:51,607148	172.31.99.168	172.31.9.118	SMB2	Tree Connect Response	<= continuous availability, scaleout, cluster	
12:27:51,763098	172.31.9.118	172.31.99.168	WITNESS	GetInterfaceList request		
12:27:51,765239	172.31.99.168	172.31.9.118	WITNESS	GetInterfaceList response, AVAILABLE Ipv4:172.31.99.166 WITNESS_IF, AVAILABLE		
12:27:51,906223	172.31.9.118	172.31.99.166	WITNESS	RegisterEx request NetName[ubcluster.w2022-17.base] IpAddress[172.31.99.168]		
12:27:51,909542	172.31.99.166	172.31.9.118	WITNESS	RegisterEx response		
12:27:51,918601	172.31.9.118	172.31.99.166	WITNESS	AsyncNotify request		
12:29:51,877453	172.31.99.166	172.31.9.118	WITNESS	AsyncNotify response, Error: WERR_TIMEOUT		
12:29:51,878346	172.31.9.118	172.31.99.166	WITNESS	AsyncNotify request		
12:31:51,919980	172.31.99.166	172.31.9.118	WITNESS	AsyncNotify response, Error: WERR_TIMEOUT		
12:31:51,920465	172.31.9.118	172.31.99.166	WITNESS	AsyncNotify request		
12:33:51,961711	172.31.99.166	172.31.9.118	WITNESS	AsyncNotify response, Error: WERR_TIMEOUT		
12:33:51,962723	172.31.9.118	172.31.99.166	WITNESS	AsyncNotify request		
12:35:51,915582	172.31.99.166	172.31.9.118	WITNESS	AsyncNotify response, Error: WERR_TIMEOUT		
12:35:51,916044	172.31.9.118	172.31.99.166	WITNESS	AsyncNotify request		

Resource-Unavailable flow

18:08:33,144233	172.31.9.118	172.31.99.167	SMB2	Negotiate Protocol Request	Client: 172.31.9.118
18:08:33,153335	172.31.99.167	172.31.9.118	SMB2	Negotiate Protocol Response	Node0: 172.31.99.166
18:08:33,154517	172.31.9.118	172.31.99.167	SMB2	Session Setup Request	Node1: 172.31.99.167
18:08:33,164231	172.31.99.167	172.31.9.118	SMB2	Session Setup Response	Node2: 172.31.99.168
18:08:33,164807	172.31.9.118	172.31.99.167	SMB2	Tree Connect Request Tree: \\ubcluster.w2022-17.base\sh	
18:08:33,165804	172.31.99.167	172.31.9.118	SMB2	Tree Connect Response	
18:08:34,143667	172.31.9.118	172.31.99.167	SMB2	Tree Connect Request Tree: \\ubcluster.w2022-17.base\IPCS	
18:08:34,144945	172.31.99.167	172.31.9.118	SMB2	Tree Connect Response	
18:08:38,255867	172.31.9.118	172.31.99.167	WITNESS	GetInterfaceList request	
18:08:38,257111	172.31.99.167	172.31.9.118	WITNESS	GetInterfaceList response, AVAILABLE Ipv4:172.31.99.166 WITNESS_IF, AVAILABLE Ipv4:172.31.99.166	
18:08:38,264767	172.31.9.118	172.31.99.166	WITNESS	RegisterEx request NetName[ubcluster.w2022-17.base] IpAddress[172.31.99.167]	
18:08:38,265795	172.31.99.166	172.31.9.118	WITNESS	RegisterEx response	
18:08:38,271850	172.31.9.118	172.31.99.166	WITNESS	AsyncNotify request	
18:10:38,328809	172.31.99.166	172.31.9.118	WITNESS	AsyncNotify response, Error: WERR_TIMEOUT	
18:10:38,329410	172.31.9.118	172.31.99.166	WITNESS	AsyncNotify request	
18:10:49,638669	172.31.99.166	172.31.9.118	WITNESS	AsyncNotify response RESOURCE_CHANGE (1 message), RESOURCE_UNAVAILABLE, 172.31.99.167[L	
18:10:49,640021	172.31.9.118	172.31.99.166	WITNESS	AsyncNotify request	
18:10:49,644707	172.31.9.118	172.31.99.166	SMB2	Negotiate Protocol Request	
18:10:49,655469	172.31.99.166	172.31.9.118	SMB2	Negotiate Protocol Response	
18:10:49,656805	172.31.9.118	172.31.99.166	SMB2	Session Setup Request	
18:10:49,668964	172.31.99.166	172.31.9.118	SMB2	Session Setup Response	
18:10:49,669895	172.31.9.118	172.31.99.166	SMB2	Tree Connect Request Tree: \\ubcluster.w2022-17.base\sh	
18:10:49,672057	172.31.99.166	172.31.9.118	SMB2	Tree Connect Response	
18:10:54,645342	172.31.99.166	172.31.9.118	WITNESS	AsyncNotify response, Error: WERR_NOT_FOUND Hack to force a re-registration	
18:10:54,646097	172.31.9.118	172.31.99.166	WITNESS	UnRegister request	
18:10:54,646673	172.31.99.166	172.31.9.118	WITNESS	UnRegister response, Error: WERR_NOT_FOUND	
18:10:54,661688	172.31.9.118	172.31.99.166	WITNESS	GetInterfaceList request	
18:10:54,662330	172.31.99.166	172.31.9.118	WITNESS	GetInterfaceList response, AVAILABLE Ipv4:172.31.99.166, UNAVAILABLE Ipv4:172.31.99.167	
18:10:54,778103	172.31.9.118	172.31.99.168	WITNESS	RegisterEx request NetName[ubcluster.w2022-17.base] IpAddress[172.31.99.166]	
18:10:54,780058	172.31.99.168	172.31.9.118	WITNESS	RegisterEx response	
18:10:54,787232	172.31.9.118	172.31.99.168	WITNESS	AsyncNotify request	

Client-Move flow

15:44:36,717268	172.31.9.118	172.31.99.167	SMB2	Negotiate Protocol Request
15:44:36,723718	172.31.99.167	172.31.9.118	SMB2	Negotiate Protocol Response
15:44:36,724414	172.31.9.118	172.31.99.167	SMB2	Session Setup Request
15:44:36,731287	172.31.99.167	172.31.9.118	SMB2	Session Setup Response
15:44:36,731763	172.31.9.118	172.31.99.167	SMB2	Tree Connect Request Tree: \\ubcluster.w2022-17.base\shm
15:44:36,732881	172.31.99.167	172.31.9.118	SMB2	Tree Connect Response
15:44:37,739894	172.31.9.118	172.31.99.167	SMB2	Tree Connect Request Tree: \\ubcluster.w2022-17.base\IPCS
15:44:37,741150	172.31.99.167	172.31.9.118	SMB2	Tree Connect Response
15:44:41,745394	172.31.9.118	172.31.99.167	WITNESS	GetInterfaceList request
15:44:41,745947	172.31.99.167	172.31.9.118	WITNESS	GetInterfaceList response, AVAILABLE Ipv4:172.31.99.166 WITNESS_IF, AVAILABLE
15:44:41,853592	172.31.9.118	172.31.99.166	WITNESS	RegisterEx request NetName[ubcluster.w2022-17.base] IpAddress[172.31.99.167]
15:44:41,855292	172.31.99.166	172.31.9.118	WITNESS	RegisterEx response
15:44:41,863502	172.31.9.118	172.31.99.166	WITNESS	AsyncNotify request
15:46:41,868076	172.31.99.166	172.31.9.118	WITNESS	AsyncNotify response, Error: WERR_TIMEOUT
15:46:41,869075	172.31.9.118	172.31.99.166	WITNESS	AsyncNotify request
15:48:41,970821	172.31.99.166	172.31.9.118	WITNESS	AsyncNotify response, Error: WERR_TIMEOUT
15:48:41,971270	172.31.9.118	172.31.99.166	WITNESS	AsyncNotify request
15:50:28,174463	172.31.99.166	172.31.9.118	WITNESS	AsyncNotify response CLIENT_MOVE (1 message) Ipv4:172.31.99.168[Long frame (12
15:50:28,175499	172.31.9.118	172.31.99.166	WITNESS	AsyncNotify request
15:50:28,176791	172.31.9.118	172.31.99.168	SMB2	Negotiate Protocol Request
15:50:28,186078	172.31.99.168	172.31.9.118	SMB2	Negotiate Protocol Response
15:50:28,186724	172.31.9.118	172.31.99.168	SMB2	Session Setup Request
15:50:28,194004	172.31.99.168	172.31.9.118	SMB2	Session Setup Response
15:50:28,194490	172.31.9.118	172.31.99.168	SMB2	Tree Connect Request Tree: \\ubcluster.w2022-17.base\shm
15:50:28,196587	172.31.99.168	172.31.9.118	SMB2	Tree Connect Response
15:50:29,196623	172.31.9.118	172.31.99.168	SMB2	Tree Connect Request Tree: \\ubcluster.w2022-17.base\IPCS
15:50:29,198861	172.31.99.168	172.31.9.118	SMB2	Tree Connect Response
15:50:33,203320	172.31.99.166	172.31.9.118	WITNESS	AsyncNotify response, Error: WERR_NOT_FOUND Hack to trigger a re-registration
15:50:33,204027	172.31.9.118	172.31.99.166	WITNESS	UnRegister request
15:50:33,204604	172.31.99.166	172.31.9.118	WITNESS	UnRegister response, Error: WERR_NOT_FOUND
15:50:33,308338	172.31.9.118	172.31.99.168	WITNESS	GetInterfaceList request
15:50:33,309865	172.31.99.168	172.31.9.118	WITNESS	GetInterfaceList response, AVAILABLE Ipv4:172.31.99.166 WITNESS_IF, AVAILABLE
15:50:33,319486	172.31.9.118	172.31.99.166	WITNESS	RegisterEx request NetName[ubcluster.w2022-17.base] IpAddress[172.31.99.168]
15:50:33,319983	172.31.99.166	172.31.9.118	WITNESS	RegisterEx response
15:50:33,325602	172.31.9.118	172.31.99.166	WITNESS	AsyncNotify request

- ▶ We had some source3/rpc_server rewrites in the last years
 - ▶ The merge to dcesrv_core.c by Samuel Cabrero
 - ▶ The samba-dcerpcd infrastructure by Volker Lendecke
- ▶ We can now have isolated service binaries
 - ▶ /usr/libexec/samba/rpcd_
 - ▶ With 'rpc start on demand helpers = no' we support ncacn_ip_tcp
- ▶ Simple async responses are possible
 - ▶ If we do not care about user impersonation

- ▶ We had some source3/rpc_server rewrites in the last years
 - ▶ The merge to dcesrv_core.c by Samuel Cabrero
 - ▶ The samba-dcerpcd infrastructure by Volker Lendecke
- ▶ We can now have isolated service binaries
 - ▶ /usr/libexec/samba/rpcd_
 - ▶ With 'rpc start on demand helpers = no' we support ncacn_ip_tcp
- ▶ Simple async responses are possible
 - ▶ If we do not care about user impersonation

- ▶ We had some source3/rpc_server rewrites in the last years
 - ▶ The merge to dcesrv_core.c by Samuel Cabrero
 - ▶ The samba-dcerpcd infrastructure by Volker Lendecke
- ▶ We can now have isolated service binaries
 - ▶ /usr/libexec/samba/rpcd_
 - ▶ With 'rpc start on demand helpers = no' we support ncacn_ip_tcp
- ▶ Simple async responses are possible
 - ▶ If we do not care about user impersonation

rpcd_witness design (Part 2)

- ▶ We had some witness service prototypes implemented in the past
 - ▶ By Gregor Beck/Stefan Metzmacher
 - ▶ By Günther Deschner/Jose A. Rivera
 - ▶ By David Disseldorp/Samuel Cabrero
- ▶ The interaction with ctdbd is important
 - ▶ But it was missing in 2 prototypes
 - ▶ And 1 prototype tried to implement too much in ctdbd itself
- ▶ Finally I came up with a very simple ctdbd change
 - ▶ It was trivial to add CTDB_SRVID_IPREALLOCATED notifications to ctdbd
- ▶ Each rpcd_witness instance just needs this:
 - ▶ Load all addresses of the whole cluster at start
 - ▶ Wait for CTDB_SRVID_IPREALLOCATED to be posted
 - ▶ Reload all addresses of the whole cluster
 - ▶ Compare the changes in the list in order to notice changes

rpcd_witness design (Part 3)

- ▶ `rpcd_witness` needs support for `ncacn_ip_tcp`
 - ▶ So it requires `'rpc start on demand helpers = no'`
 - ▶ We also register each connection with `ctdbd` to get tickle-acks
- ▶ Each `Register[Ex]()` results in a global registration
 - ▶ They are stored in `rpcd_witness_registration.tdb`
 - ▶ With the registration context/policy handle as key
 - ▶ And the `server_id` (node+pid) also in the content
- ▶ This allows 'net witness' commands to work
 - ▶ List registrations
 - ▶ Send specific administrative actions to the individual registrations
 - ▶ See later slides for more details and examples

Windows clients behave in strange ways (Part 1)

- ▶ The SMB2 Tree Connect response has flags for cluster capabilities:
 - ▶ SMB2_SHARE_CAP_CONTINUOUS_AVAILABILITY
 - ▶ SMB2_SHARE_CAP_SCALEOUT
 - ▶ SMB2_SHARE_CAP_CLUSTER
 - ▶ SMB2_SHARE_CAP_ASYMMETRIC
- ▶ SMB2_SHARE_CAP_CLUSTER:
 - ▶ This is the indication the [MS-SWN] service runs on the server
 - ▶ And the client should make use of it when using the connected share
 - ▶ Sadly only effective together with SMB2_SHARE_CAP_CONTINUOUS_AVAILABILITY
- ▶ SMB2_SHARE_CAP_SCALEOUT:
 - ▶ Means the cluster can have more that one active node at a time

Windows clients behave in strange ways (Part 1)

- ▶ The SMB2 Tree Connect response has flags for cluster capabilities:
 - ▶ SMB2_SHARE_CAP_CONTINUOUS_AVAILABILITY
 - ▶ SMB2_SHARE_CAP_SCALEOUT
 - ▶ SMB2_SHARE_CAP_CLUSTER
 - ▶ SMB2_SHARE_CAP_ASYMMETRIC
- ▶ SMB2_SHARE_CAP_CLUSTER:
 - ▶ This is the indication the [MS-SWN] service runs on the server
 - ▶ And the client should make use of it when using the connected share
 - ▶ Sadly only effective together with SMB2_SHARE_CAP_CONTINUOUS_AVAILABILITY
- ▶ SMB2_SHARE_CAP_SCALEOUT:
 - ▶ Means the cluster can have more that one active node at a time

Windows clients behave in strange ways (Part 1)

- ▶ The SMB2 Tree Connect response has flags for cluster capabilities:
 - ▶ SMB2_SHARE_CAP_CONTINUOUS_AVAILABILITY
 - ▶ SMB2_SHARE_CAP_SCALEOUT
 - ▶ SMB2_SHARE_CAP_CLUSTER
 - ▶ SMB2_SHARE_CAP_ASYMMETRIC
- ▶ SMB2_SHARE_CAP_CLUSTER:
 - ▶ This is the indication the [MS-SWN] service runs on the server
 - ▶ And the client should make use of it when using the connected share
 - ▶ Sadly only effective together with SMB2_SHARE_CAP_CONTINUOUS_AVAILABILITY
- ▶ SMB2_SHARE_CAP_SCALEOUT:
 - ▶ Means the cluster can have more that one active node at a time

Windows clients behave in strange ways (Part 2)

- ▶ **SMB2_SHARE_CAP_CONTINUOUS_AVAILABILITY:**
 - ▶ This indicates that the share is always available
 - ▶ The client should try to reconnect (maybe to other nodes) fast
 - ▶ Windows clients also use this as trigger to request presistent handles
 - ▶ Even is the server does not provide SMB2_CAP_PERSISTENT_HANDLES
 - ▶ Each open generates a warning in the client event log
- ▶ **SMB2_SHARE_CAP_ASYMMETRIC:**
 - ▶ This is used to indicate that a share is attached to a disk owner
 - ▶ Other nodes act as proxy.
 - ▶ It means the client uses separate set of connections for the share
 - ▶ The client might connect to a different cluster node
 - ▶ And provides a share name for RegisterEx()

Windows clients behave in strange ways (Part 2)

- ▶ **SMB2_SHARE_CAP_CONTINUOUS_AVAILABILITY:**
 - ▶ This indicates that the share is always available
 - ▶ The client should try to reconnect (maybe to other nodes) fast
 - ▶ Windows clients also use this as trigger to request presistent handles
 - ▶ Even is the server does not provide SMB2_CAP_PERSISTENT_HANDLES
 - ▶ Each open generates a warning in the client event log
- ▶ **SMB2_SHARE_CAP_ASYMMETRIC:**
 - ▶ This is used to indicate that a share is attached to a disk owner
 - ▶ Other nodes act as proxy.
 - ▶ It means the client uses separate set of connections for the share
 - ▶ The client might connect to a different cluster node
 - ▶ And provides a share name for RegisterEx()

Windows clients behave in strange ways (Part 3)

- ▶ After an AsyncNotify response there is no re-registration
 - ▶ A Windows client reacts on a RESOURCE_CHANGE, CLIENT_MOVE, SHARE_MOVE.
 - ▶ It reconnects the SMB3 connection if required
 - ▶ But it does not call Register[Ex]() for the new connection
- ▶ We use a trick in order to force a re-registration
 - ▶ 5 seconds after a RESOURCE_CHANGE, CLIENT_MOVE, SHARE_MOVE.
 - ▶ we return AsyncNotify with STATUS_NOT_FOUND
 - ▶ This triggers a re-registration

Windows clients behave in strange ways (Part 3)

- ▶ After an AsyncNotify response there is no re-registration
 - ▶ A Windows client reacts on a RESOURCE_CHANGE, CLIENT_MOVE, SHARE_MOVE.
 - ▶ It reconnects the SMB3 connection if required
 - ▶ But it does not call Register[Ex]() for the new connection
- ▶ We use a trick in order to force a re-registration
 - ▶ 5 seconds after a RESOURCE_CHANGE, CLIENT_MOVE, SHARE_MOVE.
 - ▶ we return AsyncNotify with STATUS_NOT_FOUND
 - ▶ This triggers a re-registration

Basic smb.conf options for rpcd_witness

net conf list output:

```
[global]
  netbios name = ubcluster
  idmap config * : backend = autorid
  idmap config * : range = 1000000-1999999
  security = ADS
  workgroup = W2022-L7
  realm = W2022-L7.BASE
  rpc start on demand helpers = no
  smb3 share cap:continuous availability = yes

[shm]
  path = /dev/shm
  read only = no
```

- ▶ There is a 47.samba-dcerpcd script for ctdbd
 - ▶ 'ctdb event script enable legacy 47.samba-dcerpcd'
 - ▶ This tries to start the samba-dcerpcd (systemd service)
 - ▶ This is needed for 'rpc start on demand helpers = no'

- ▶ net witness list
 - ▶ List witness registrations from rpcd_witness_registration.tdb
- ▶ net witness client-move
 - ▶ Generate client move notifications for witness registrations to a new ip or node
- ▶ net witness share-move
 - ▶ Generate share move notifications for witness registrations to a new ip or node
- ▶ net witness force-unregister
 - ▶ Force unregistrations for witness registrations
- ▶ net witness force-response
 - ▶ Force an AsyncNotify response based on json input (mostly for testing)

- ▶ net witness list
 - ▶ List witness registrations from rpcd_witness_registration.tdb
- ▶ net witness client-move
 - ▶ Generate client move notifications for witness registrations to a new ip or node
- ▶ net witness share-move
 - ▶ Generate share move notifications for witness registrations to a new ip or node
- ▶ net witness force-unregister
 - ▶ Force unregistrations for witness registrations
- ▶ net witness force-response
 - ▶ Force an AsyncNotify response based on json input (mostly for testing)

- ▶ net witness list
 - ▶ List witness registrations from rpcd_witness_registration.tdb
- ▶ net witness client-move
 - ▶ Generate client move notifications for witness registrations to a new ip or node
- ▶ net witness share-move
 - ▶ Generate share move notifications for witness registrations to a new ip or node
- ▶ net witness force-unregister
 - ▶ Force unregistrations for witness registrations
- ▶ net witness force-response
 - ▶ Force an AsyncNotify response based on json input (mostly for testing)

- ▶ net witness list
 - ▶ List witness registrations from rpcd_witness_registration.tdb
- ▶ net witness client-move
 - ▶ Generate client move notifications for witness registrations to a new ip or node
- ▶ net witness share-move
 - ▶ Generate share move notifications for witness registrations to a new ip or node
- ▶ net witness force-unregister
 - ▶ Force unregistrations for witness registrations
- ▶ net witness force-response
 - ▶ Force an AsyncNotify response based on json input (mostly for testing)

net witness commands

- ▶ net witness list
 - ▶ List witness registrations from rpcd_witness_registration.tdb
- ▶ net witness client-move
 - ▶ Generate client move notifications for witness registrations to a new ip or node
- ▶ net witness share-move
 - ▶ Generate share move notifications for witness registrations to a new ip or node
- ▶ net witness force-unregister
 - ▶ Force unregistrations for witness registrations
- ▶ net witness force-response
 - ▶ Force an AsyncNotify response based on json input (mostly for testing)

net witness list example

```
root@ubl704-166:~# net witness list
Registration-UUID:          NetName          ShareName          IpAddress          ClientComputerName
-----
c10b4d0b-758a-4918-b1fa-3791e6c4465c ubcluster.w2022-l7.base ' 172.31.99.167 w2022-118.w2022-l7.base
root@ubl704-166:~# net witness list --json --witness-registration=c10b4d0b-758a-4918-b1fa-3791e6c4465c | jq '.registrations'
{
  "c10b4d0b-758a-4918-b1fa-3791e6c4465c": {
    "version": "0x00020000",
    "net_name": "ubcluster.w2022-l7.base",
    "share_name": null,
    "ip_address": "172.31.99.167",
    "client_computer_name": "w2022-118.w2022-l7.base",
    "flags": {
      "WITNESS_REGISTER_IP_NOTIFICATION": false,
      "int": 0,
      "hex": "0x00000000"
    },
    "timeout": 120,
    "context_handle": {
      "handle_type": 1,
      "uuid": "c10b4d0b-758a-4918-b1fa-3791e6c4465c"
    },
    "server_id": {
      "pid": 25488,
      "task_id": 0,
      "vnn": 0,
      "unique_id": 1778832427806360300
    },
    "auth": {
      "account_name": "W2022-1185",
      "domain_name": "W2022-L7",
      "account_sid": "S-1-5-21-133451344-1126667713-3548050118-1000"
    },
    "connection": {
      "local_address": "ipv4:172.31.99.166:49154",
      "remote_address": "ipv4:172.31.9.118:64990"
    },
    "registration_time": "2024-04-15T14:23:51.526821+0200"
  }
}
```


net witness client-move examples

Example 1: with given registration id

```
root@ub1704-166:~# net witness client-move --witness-registration=c10b4d0b-758a-4918-b1fa-3791e6c4465c --witness-new-node=0
CLIENT_MOVE_TO_NODE: 0
Registration-UUID:          NetName          ShareName          IPAddress          ClientComputerName
-----
c10b4d0b-758a-4918-b1fa-3791e6c4465c ubcluster.w2022-l7.base ''          172.31.99.167     w2022-118.w2022-l7.base
root@ub1704-166:~# net witness list
Registration-UUID:          NetName          ShareName          IPAddress          ClientComputerName
-----
e52a060b-948b-4499-a592-1f42b90a5a5f ubcluster.w2022-l7.base ''          172.31.99.166     w2022-118.w2022-l7.base
```

Example 2: apply to all

```
root@ub1704-166:~# net witness list
Registration-UUID:          NetName          ShareName          IPAddress          ClientComputerName
-----
b217fc5d-a0c7-44a9-90f9-13228365bd21 ubcluster.w2022-l7.base ''          172.31.99.167     w2022-118.w2022-l7.base
root@ub1704-166:~# net witness client-move --witness-apply-to-all --witness-new-node=2
CLIENT_MOVE_TO_NODE: 2
Registration-UUID:          NetName          ShareName          IPAddress          ClientComputerName
-----
b217fc5d-a0c7-44a9-90f9-13228365bd21 ubcluster.w2022-l7.base ''          172.31.99.167     w2022-118.w2022-l7.base
root@ub1704-166:~#
root@ub1704-166:~# net witness list
Registration-UUID:          NetName          ShareName          IPAddress          ClientComputerName
-----
5b652b6d-4a60-4df3-9e3f-d893cf875552 ubcluster.w2022-l7.base ''          172.31.99.168     w2022-118.w2022-l7.base
```

- ▶ Samba 4.20.0 contains all changes
- ▶ We should hope that Windows clients get a fix
 - ▶ So that SMB2_SHARE_CAP_CONTINUOUS_AVAILABILITY without SMB2_CAP_PERSISTENT_HANDLES does not flood the clients event log

- ▶ Samba 4.20.0 contains all changes
- ▶ We should hope that Windows clients get a fix
 - ▶ So that SMB2_SHARE_CAP_CONTINUOUS_AVAILABILITY without SMB2_CAP_PERSISTENT_HANDLES does not flood the clients event log

Questions? Feedback!

- ▶ Stefan Metzmacher, metze@samba.org
- ▶ <https://www.sernet.com>
- ▶ <https://samba.plus>

Slides: <https://samba.org/~metze/presentations/2024/SDC/>