Installing AdRem Software's NetCrunch 9.3.3 In a Microsoft Failover Cluster

Jacob Van Vliet | M.C. Dean | 16 August 2017

Contents

Revision History

Version	Date	Modified By	Sections Modified	Description
1.0	14 April 2017	J. Van Vliet	Entire Document	Initial Writing
1.1	16 August 2017	J. Van Vliet	Entire Document	Added Screenshots, updated procedure

Executive Summary

NetCrunch is a server monitoring and analysis tool developed by AdRem Software, a Krakow, Poland-based company with offices in New York, NY and Austin, TX.

System Health and Monitoring tools have become a vital part of a company's infrastructure. Being able to know when any system or component is down, or performance may be degraded, is key to ensuring availability of services. As of 16 August 2017 NetCrunch only supports running on a single Windows Server instance. When Microsoft's "Patch Tuesday" comes, the server running NetCrunch, at some point, must be rebooted to install security updates. For most companies this downtime is okay; a vast majority of security updates won't affect the performance of the system and application.

However, customers exist where the infrastructure is so large, they cannot risk missing notifications for any device being degraded or down. The negative outcomes of a potentially losing a monitoring server due to a security update are large enough to warrant a more robust, highly available solution.

AdRem Software supports High Availability for NetCrunch 9.3.3 through VMware's Fault-Tolerant (FT) system within vSphere. VMware FT runs 2 instances of a FT Virtual machine (VM) on separate hosts; these 2 FT VM's are in "lockstep" with one another, with one VM being the primary VM. Should something happen to the host the primary VM is running on, FT immediately fails all traffic over to the secondary VM/host. Often times, this is instantaneous (within 1 second), and without any impacts to service.

As reliable as VMware FT is, not all environments run VMware vSphere. Should a company looking for a highly-available NetCrunch instance not be running VMware, or they do not have a license (or enough licenses) to cover Fault Tolerance for their environment, they would then be out of luck.

Purpose

The purpose of this document is to record the installation and configuration steps required to install NetCrunch 9.3.3 using Microsoft Failover Clustering on Windows Server 2016. This is to illustrate how a company that may be running a virtualization environment other than VMware vSphere, or that may not have licenses for Fault Tolerance within vSphere, can still maintain a highly-available application, being able to run on one of multiple servers, and not risk losing an entire NetCrunch installation when a server goes down. The same process will work on Windows Server 2012 R2.

Background

This document was created using virtual machines running in a VMware vSphere/ESXi 5.5 environment running on an IBM Flex chassis. All hosts have 32 logical cores at 2.20 GHz (2x Intel Xeon E5-2660 8c/16t @ 2.20GHz per core), and 192GB of RAM. All hosts are connected to Ethernet and Fiber Channel networks via dual 10Gbps Converged Network Adapters (CNAs) in each blade. For this particular demonstration we will be using Shared Virtual Disks (VMDKs) inside vSphere, however any method of shared disks (iSCSI, Fibre Channel, Raw Device Mappings, External SAS, etc.) will work.

Prerequisites

At a minimum, you will need:

- 2 virtual machines running Windows Server 2012 R2 or Server 2016 configured as follows-
    - NetCrunch Servers-
        - Minimum 4 cores
        - Minimum 8 GB RAM
        - Minimum 80GB disk space
        - Minimum 1Gbps network
- NetCrunch 9.3.3, downloaded from Adrem Software's website (here)

This document assumes you have Windows installed on the VMs already, and all security updates and patches from Microsoft have been installed. It also assumes all VMs are bound to an Active Directory Domain and can communicate with each other.

Installing NetCrunch 9.3.3 in a Server 2016 Failover Cluster

Configuring the Virtual Machines

1. Create 2 virtual machines with the recommended minimum settings (from above).
2. For VMware environments, follow the steps from this page to configure the disk sharing. For non-VMware environments, follow whatever steps are required for presenting a single disk/LUN to multiple servers. This may include configuring your SAN to allow the new servers to connect and mount the storage.

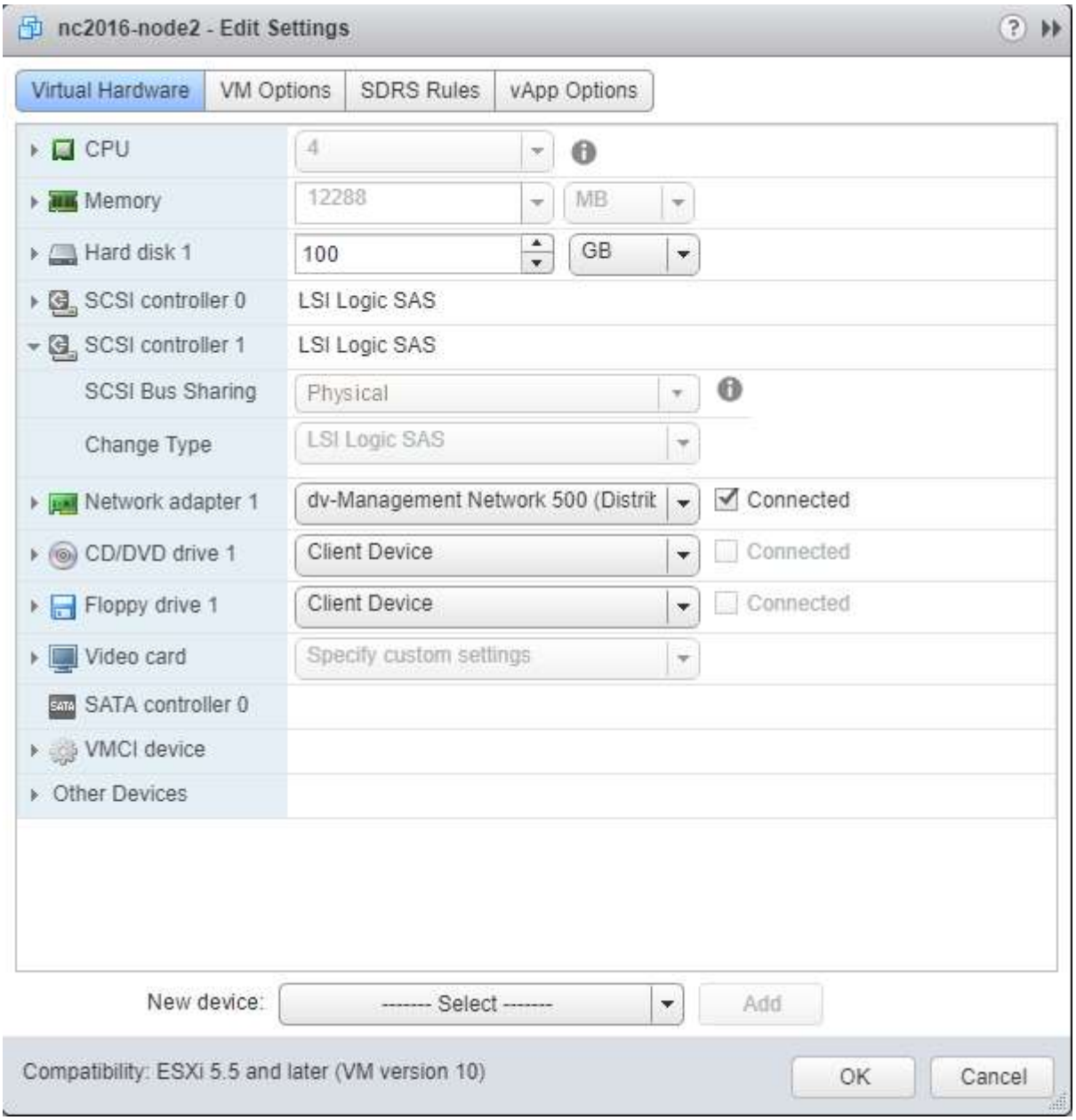

Figure 1 - VM Settings

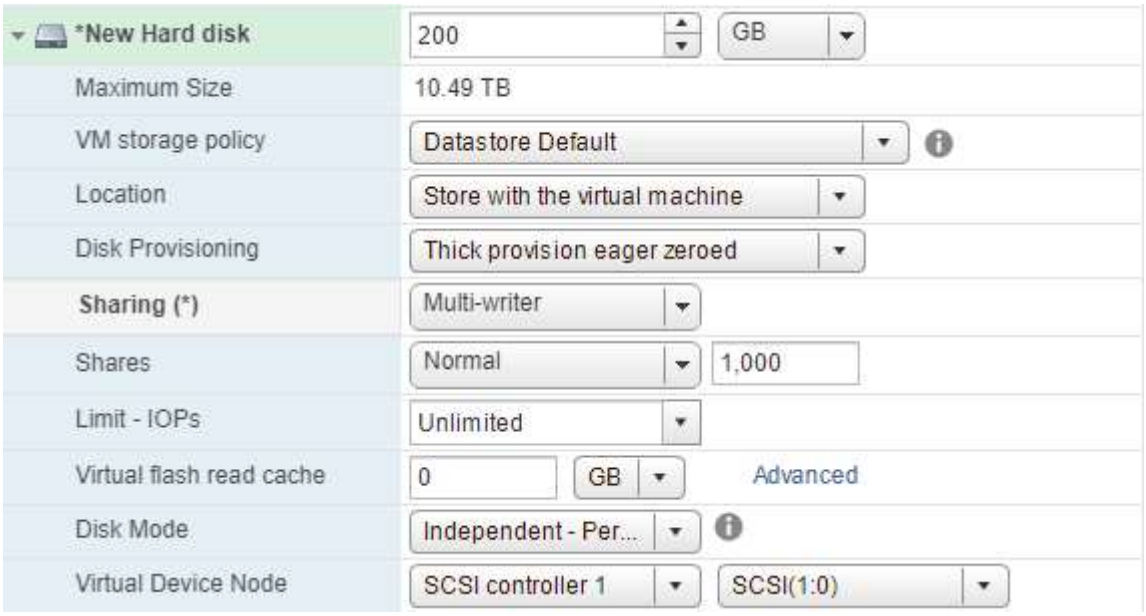

Figure 2 - Shared Disk Settings on VM1

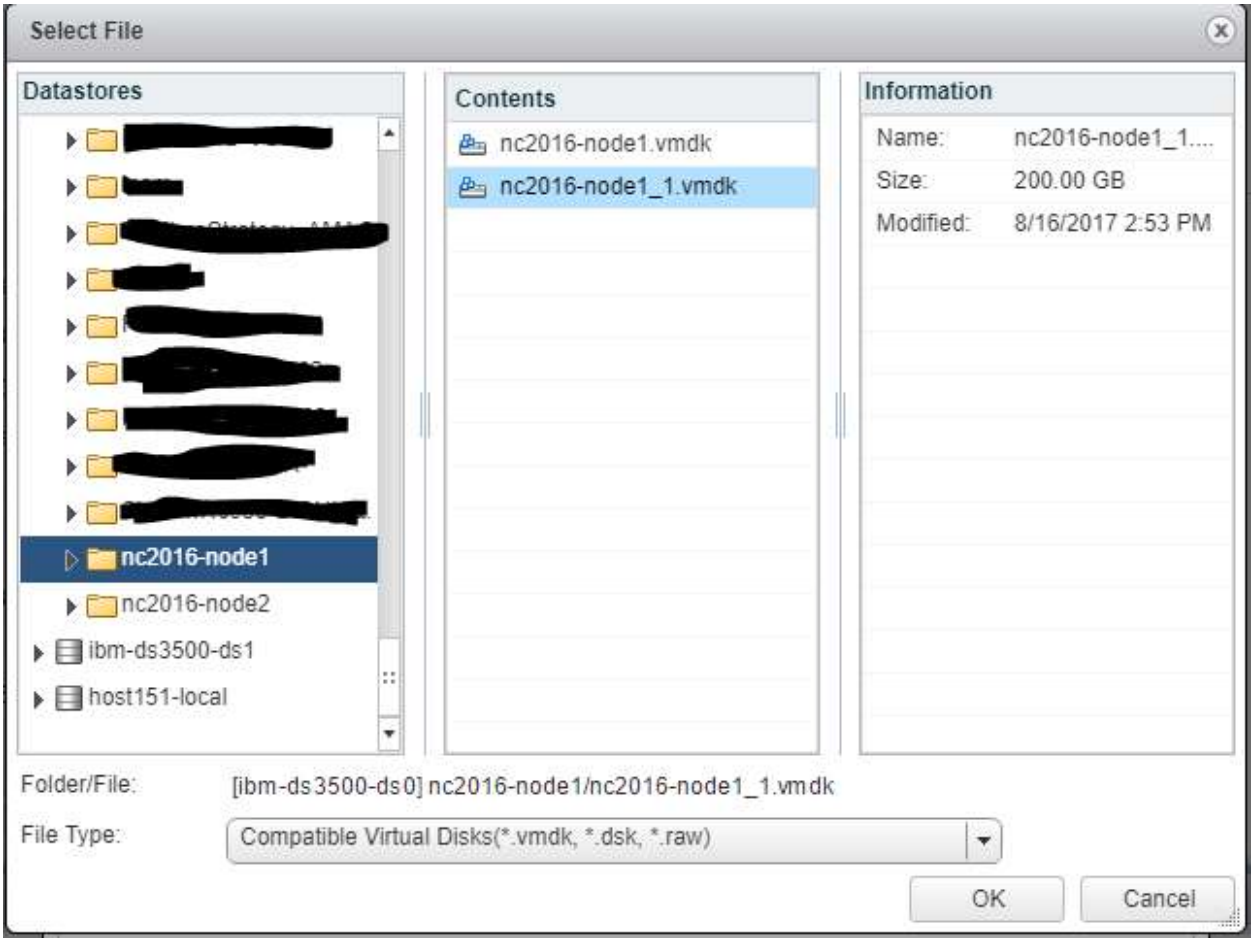

Figure 3 - Adding the shared disk to the 2nd VM

Configuring the NetCrunch Servers

Run each step below on each NetCrunch server.

1. Install the Failover Clustering server role and any dependencies, by going to Server Manager -> Manage -> Add Roles and Features.

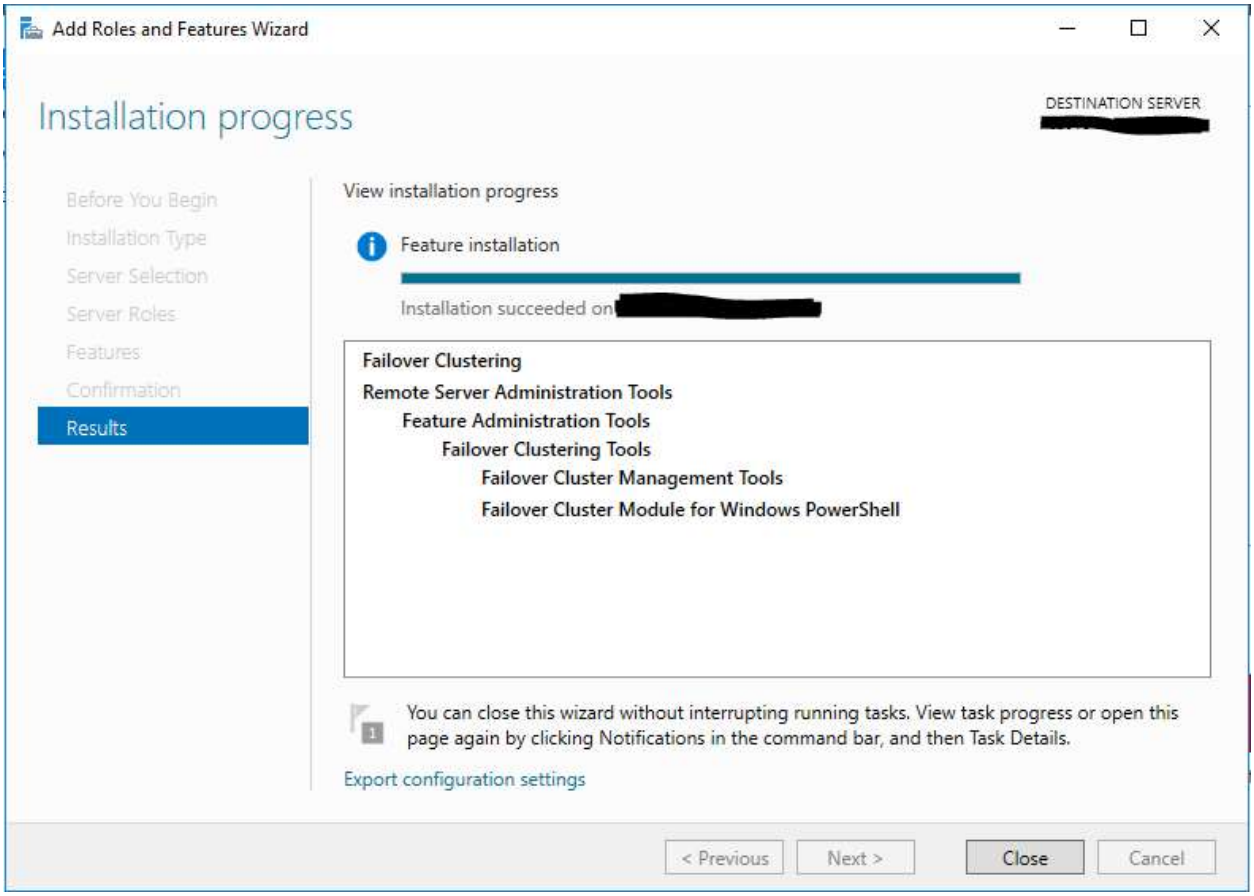

Figure 4 - Installing the Failover Clustering feature

2. Open Disk Management. Find the disks you created and Initialize the disks.
3. ON ONE SERVER ONLY – Format the disks as NTFS and give it a name/letter.

Creating the Failover Cluster

1. Once the Failover Clustering feature installation has completed, open Failover Cluster Manager by going to Server Manager -> Tools -> Failover Cluster Manager.
2. Select “Create Cluster” from the Actions menu on the right hand side of the window.

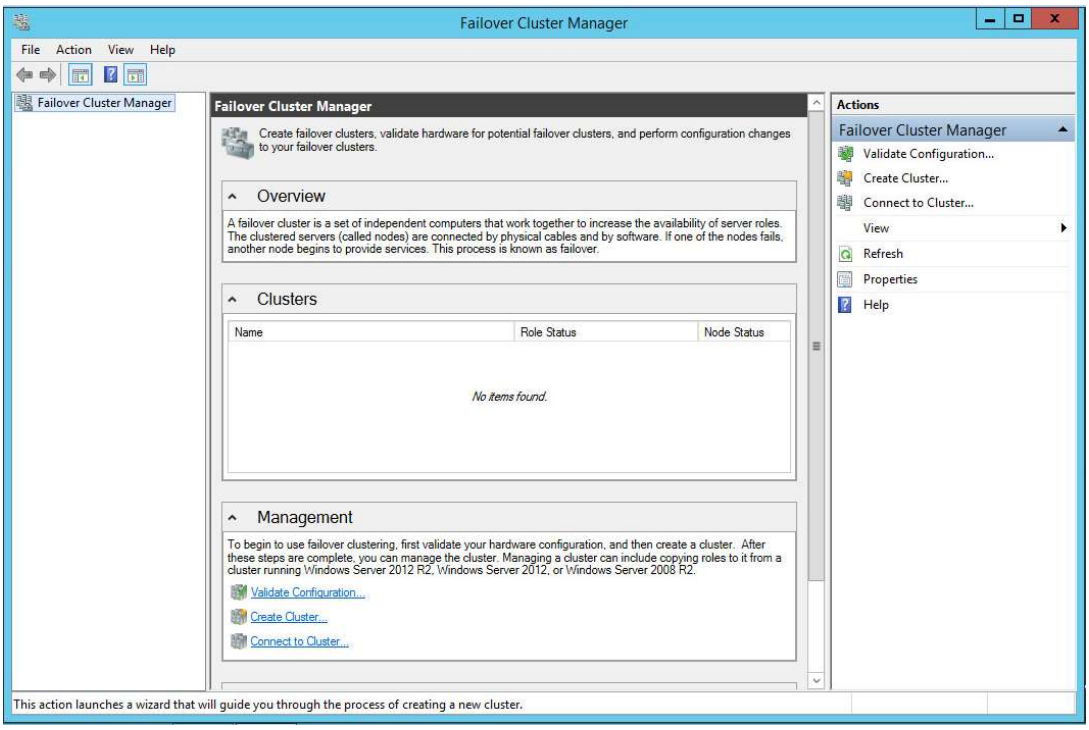

Figure 5 - Failover Cluster Manager

3. At the Select Servers page in the wizard, click Browse and add both of your NetCrunch servers.

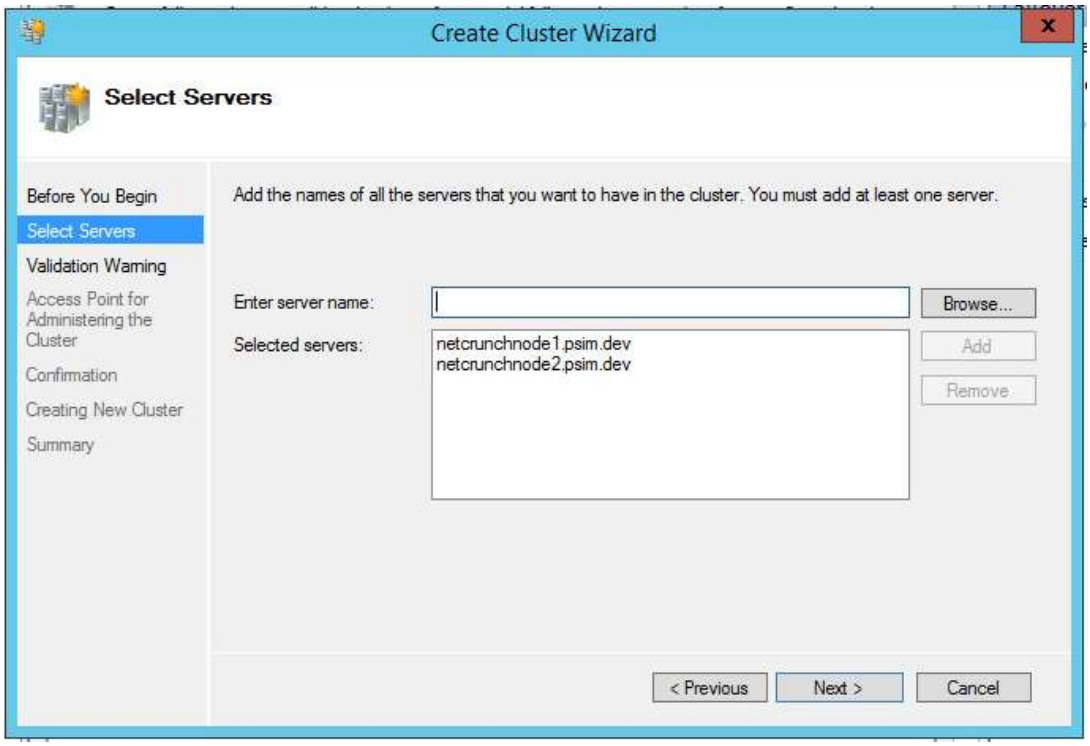

Figure 6 - Creating the Failover Cluster

4. When prompted, run the Configuration Validation tests. Ensure that all tests pass before continuing past this point.

Figure 7 - Running Validation Tests

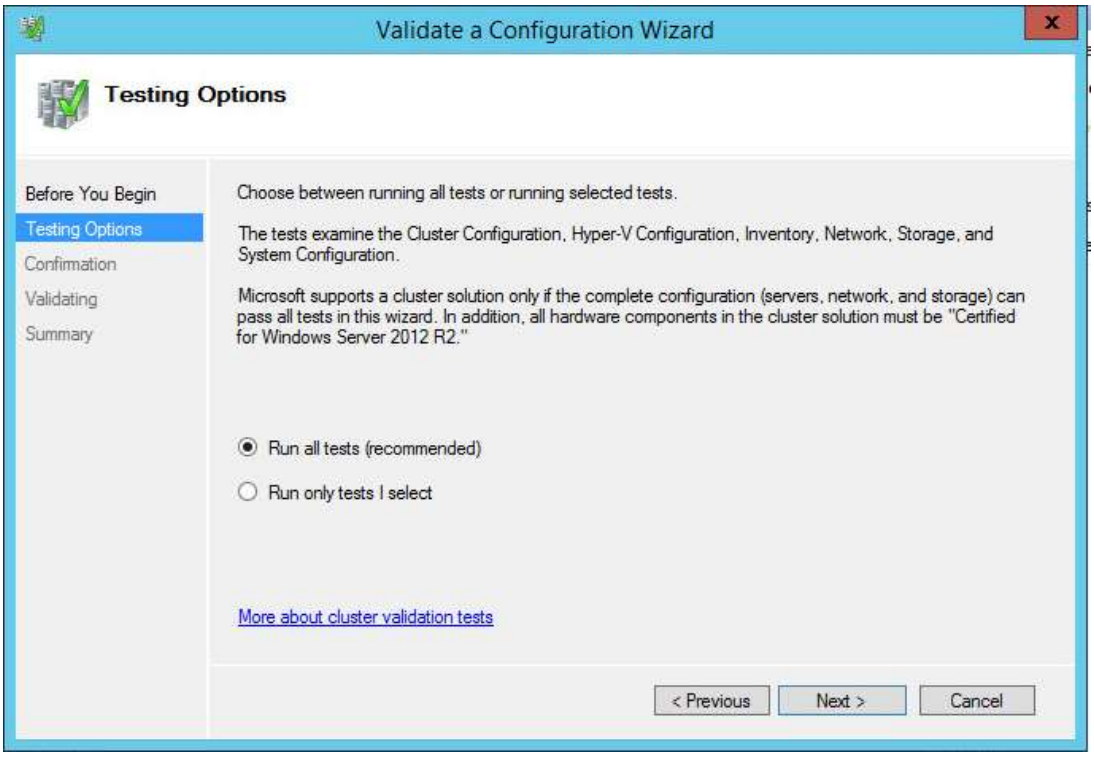

Figure 8 - Running Validation Tests

5. Enter the name and IP address you would like to connect to this cluster with. An object in AD will be created with this name, as well as a DNS record for the IP address entered.
6. Ensure the "Add All Available Storage to the Cluster" box is checked.
7. Complete the wizard and wait while the cluster is formed.
8. When the wizard has completed, open the "Configure Cluster Quorum Wizard" by going to More Actions (in the Actions menu) -> Configure Cluster Quorum.
9. In the Cluster Quorum Wizard, select "Select the Quorum witness" and press Next.
10. Select "Configure a File Share Witness". Press Next.
11. In the File Share Path, enter the UNC path to an available file share that can be used as cluster quorum (this file share only consumes 2-3MB at most). Press next and complete the wizard.
12. In Failover Cluster Manager, go to the Disks page (under Storage) and ensure the shared disk is visible. Right-click the disk and select "Add to Cluster Shared Volumes". To now access this disk, browse to C:\ClusterStorage and select the folder. This is an alias that points to the shared storage.

Installing NetCrunch

1. On the first NetCrunch server, run the NetCrunch installer previously downloaded.
2. On the Installation directory page, change the Installation Directory to be C:\ClusterStorage\$*Volume1*\Program Files…, where $Volume1 is the name of the folder/volume that appears in C:\ClusterStorage.

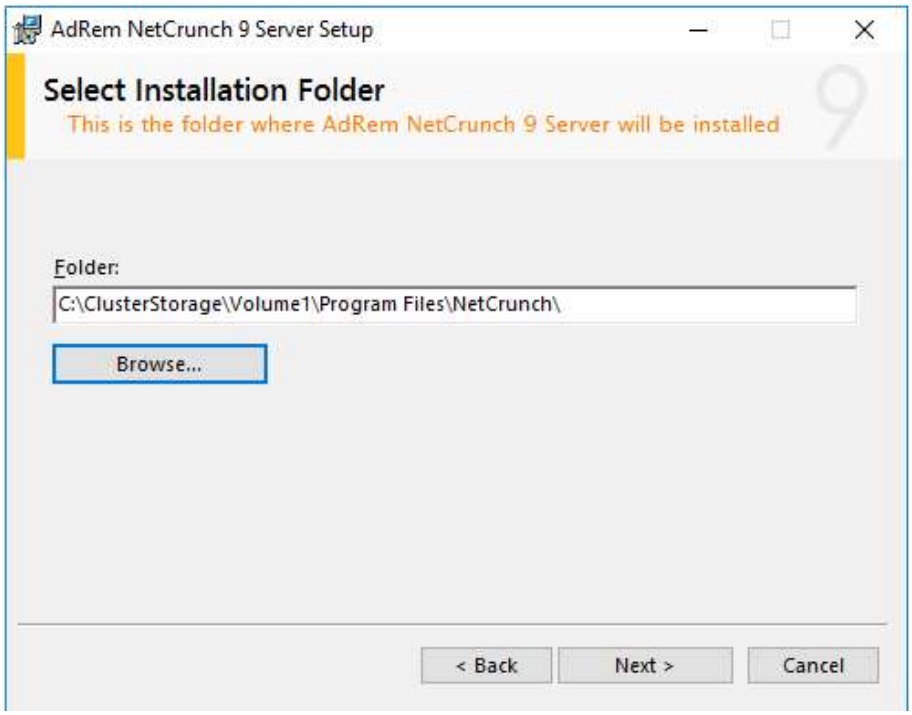

Figure 9 - NetCrunch Installation Folder

3. On the Program Data directory page, change the Program Data Directory to be C:\ClusterStorage\$*Volume1*\ProgramData…, where $Volume1 is the name of the folder/volume that appears in C:\ClusterStorage.

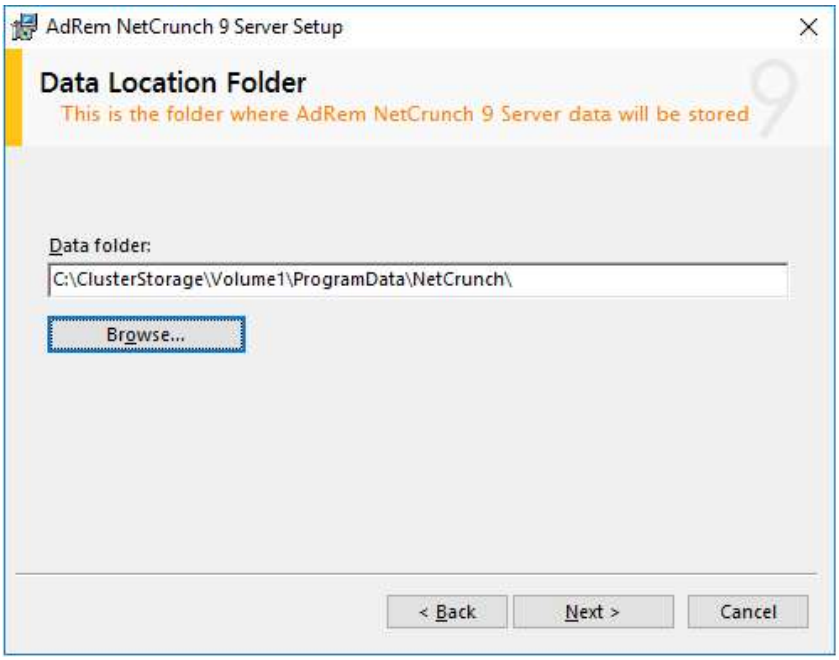

Figure 10 - NetCrunch Data Folder

4. Continue through the installation as if it were a regular install.

5. Once the installation has completed, uncheck the boxes for “Run NetCrunch Console” and “Open NetCrunch Getting Started Page” and click finish.
6. Run Services.msc and stop all NetCrunch services.

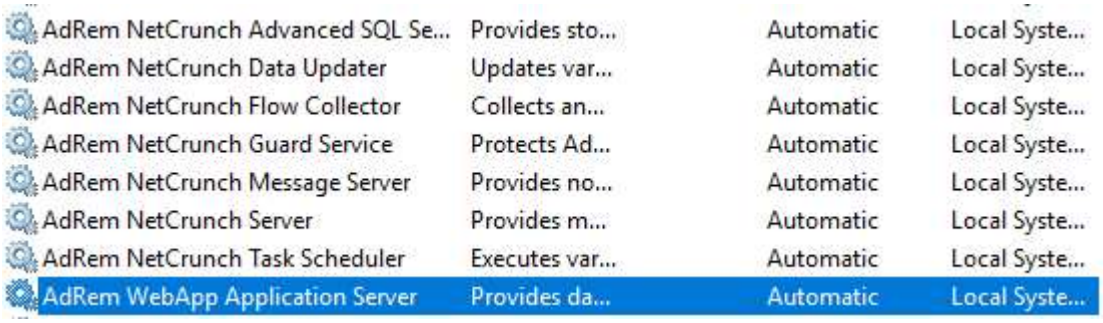

Figure 11 - NetCrunch Services

7. Repeat steps 1-4 on the second server, ensuring you set the EXACT SAME SETTINGS as you did on the primary server. Once the installation has finished, open the NetCrunch Console and verify the application is running successfully on the second server.
8. If the application runs on the second server, run Services.msc and stop all NetCrunch services on the second server.
9. Restart NetCrunch services on the primary server and attempt to open the NetCrunch console.
10. Test the application by creating a new empty Atlas and adding a node or two in to monitor. Once added, stop all NetCrunch services on the primary server, start the on the secondary, and try logging in. You should be able to see the nodes you added in.
11. After some testing, stop **all** NetCrunch services on all servers.

Making NetCrunch Highly Available

1. Open Failover Cluster Manager, and connect to your Failover Cluster.
2. In the tree on the left, select Roles.
3. In the Actions menu on the right side of the Window, click Configure Role.
4. On the Select Role page, select Other Server.

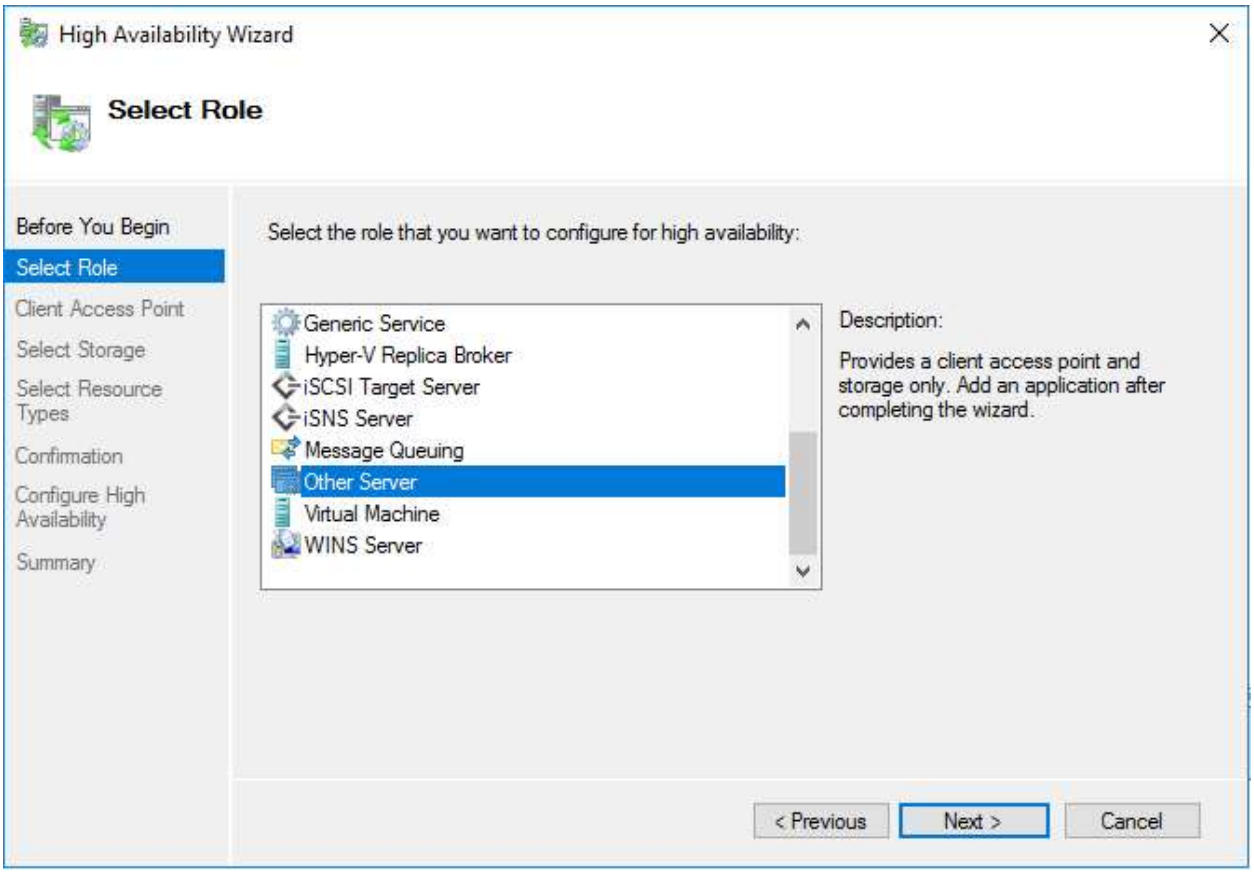

Figure 12 - Creating the Role

5. Enter a DNS name and IP address clients will connect to, and complete the wizard.

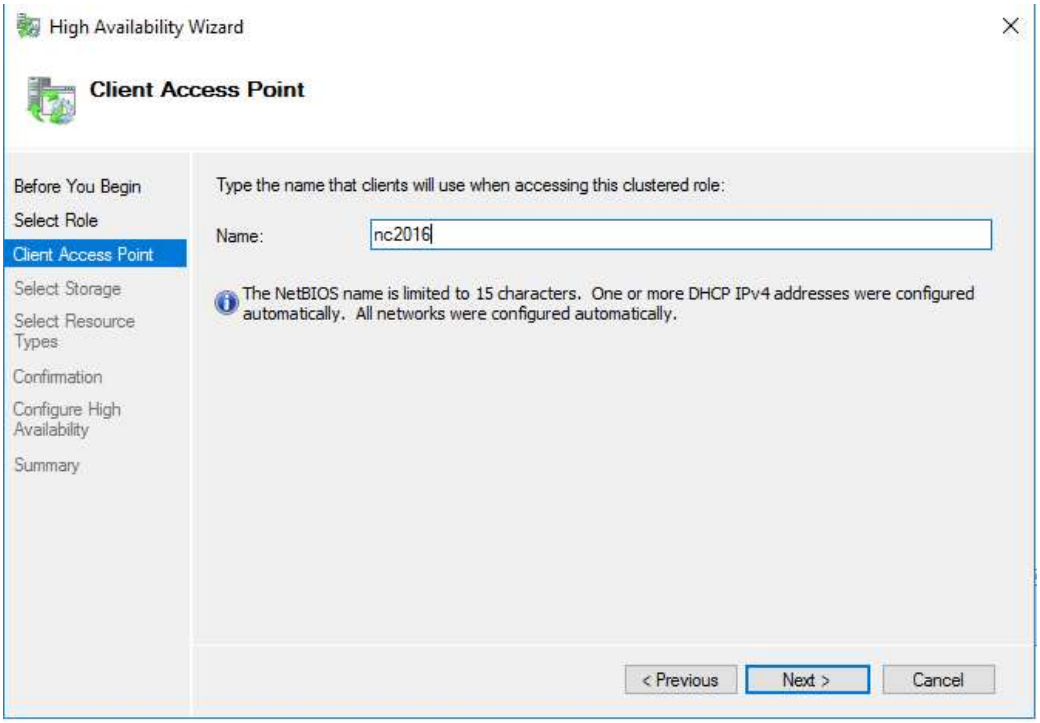

Figure 13 - Naming the Client Access Point

6. Complete the wizard without selecting other options.

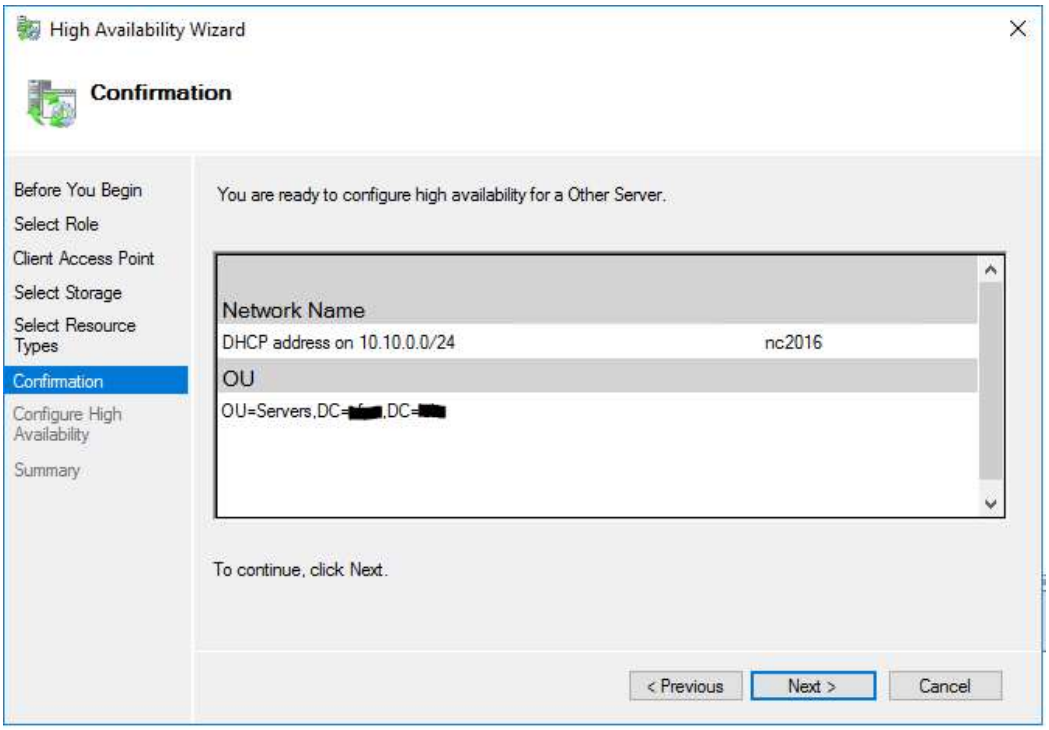

Figure 14 - Ready to create the Role

7. Once the role has been created, right click the role and select Add Resource -> Generic Service.

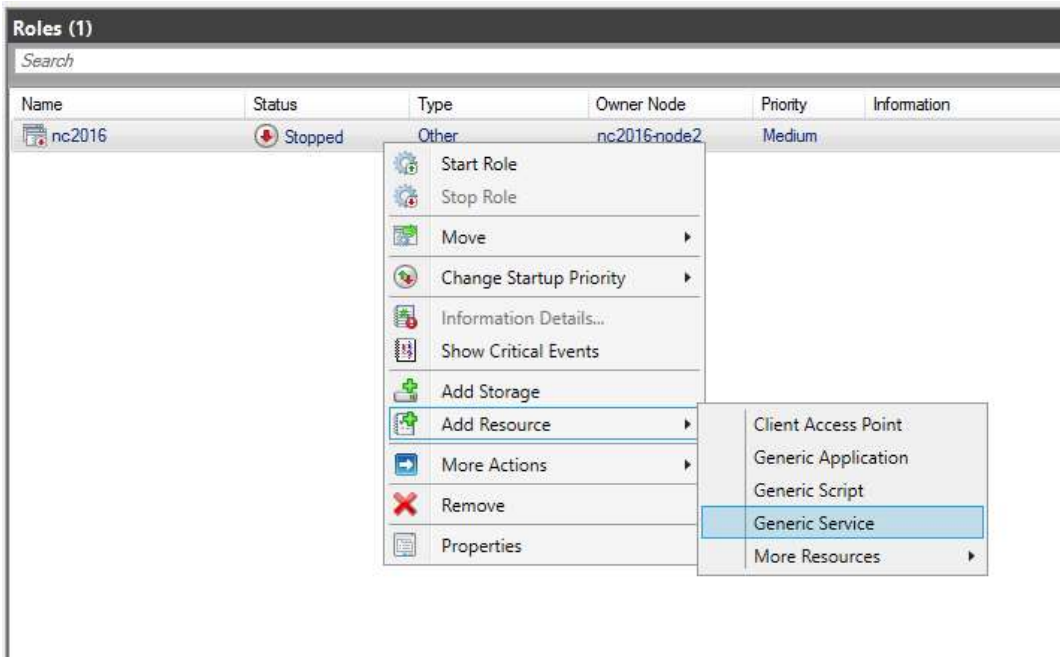

Figure 15 - Add a Generic Service

8. Select the AdRem NetCrunch Advanced SQL Server service from the list, and complete the wizard.

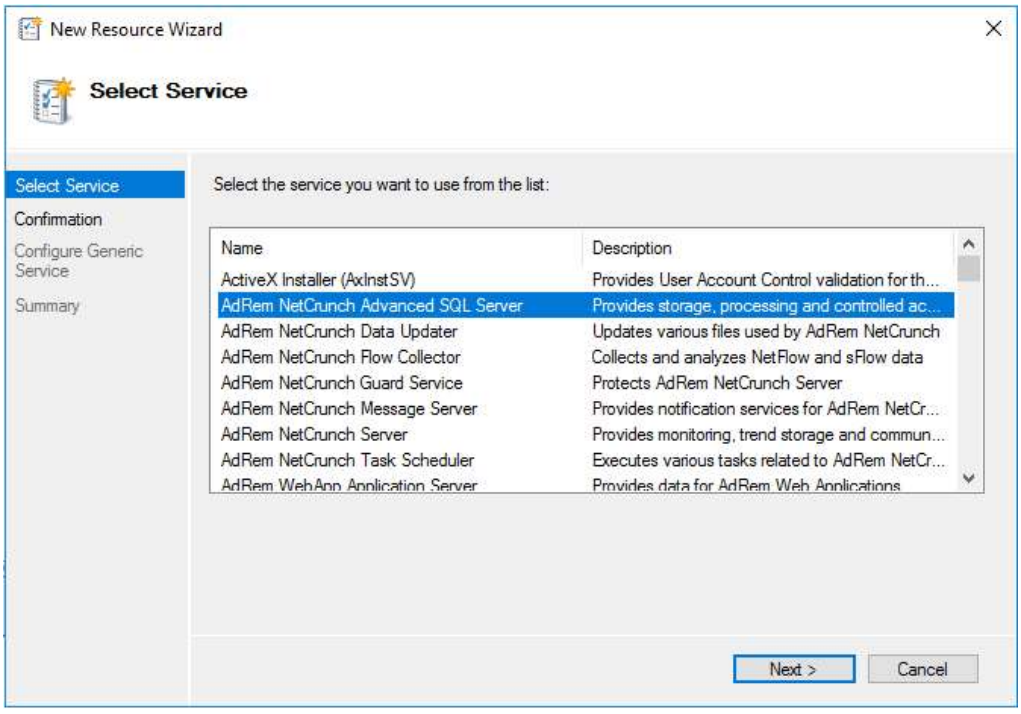

Figure 16 - Add a Generic Service

9. Repeat steps 6-7 for each of the AdRem NetCrunch services, adding the NetCrunch Server service last.

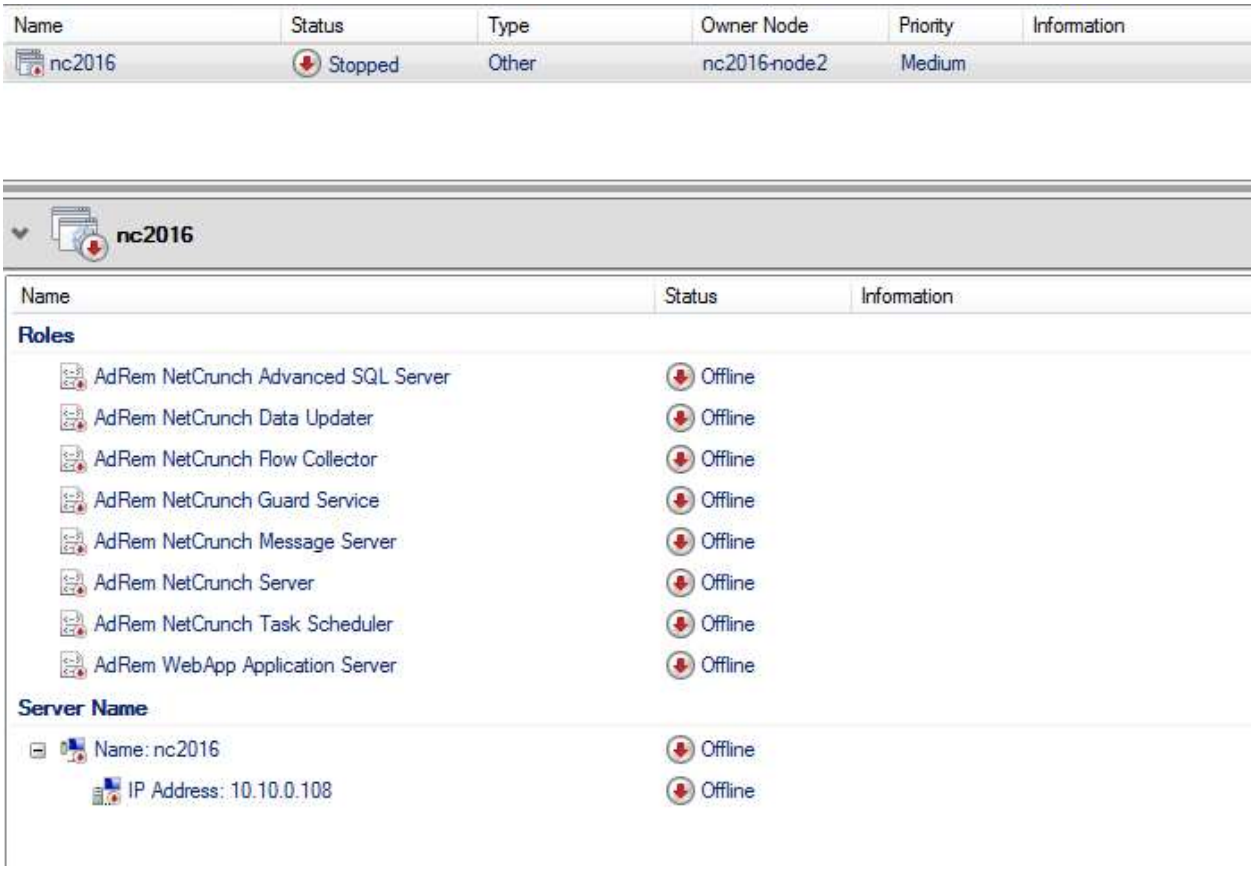

Figure 17 - All NetCrunch Services Added

10. Once all the services have been added, right-click the AdRem NetCrunch Server role and select Properties.
11. On the Dependencies tab of the NetCrunch Server service, add all the other NetCrunch services as dependencies for the NetCrunch Server service.

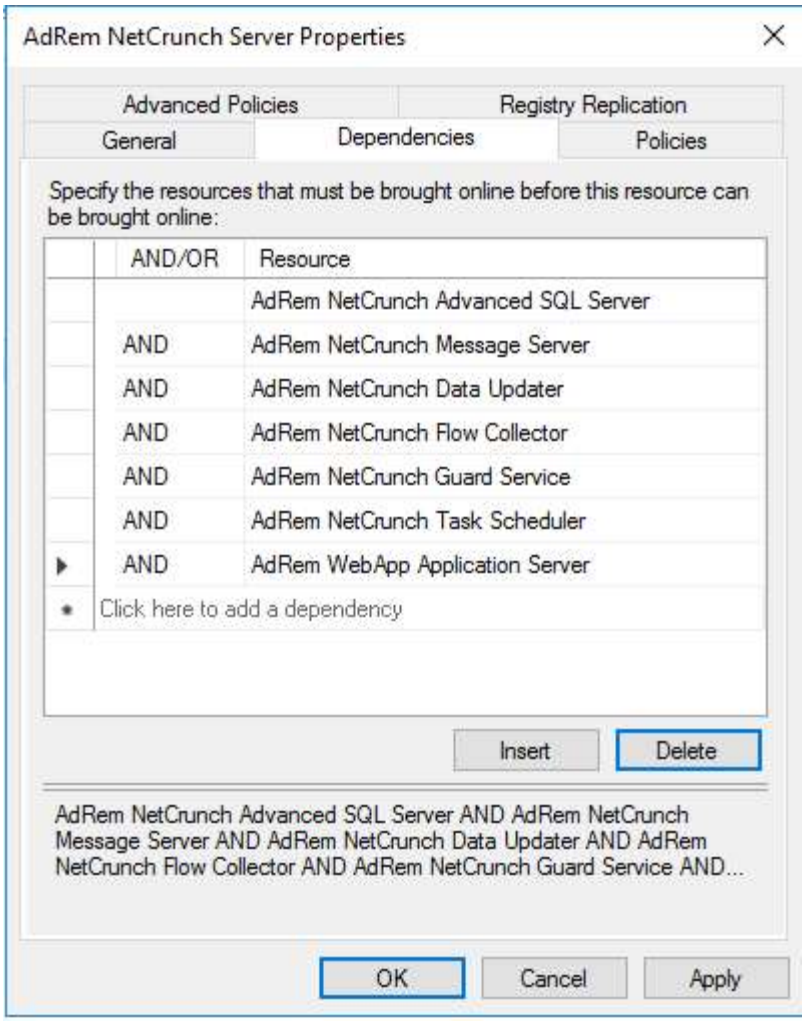

Figure 18 - Configuring Service Dependencies

12. Once all the dependencies have been added, click Apply. Right click the "nc2016" role from the top of the window and select Start Role.

Testing

1. In the System Tray (where the clock is), right-click the NetCrunch icon and select Server Connections.
2. Select the Local connection and choose Properties.
3. In the IP Address/Name field at the top, enter the Cluster Name (from "Making NetCrunch Highly Available", step 5) and press OK.

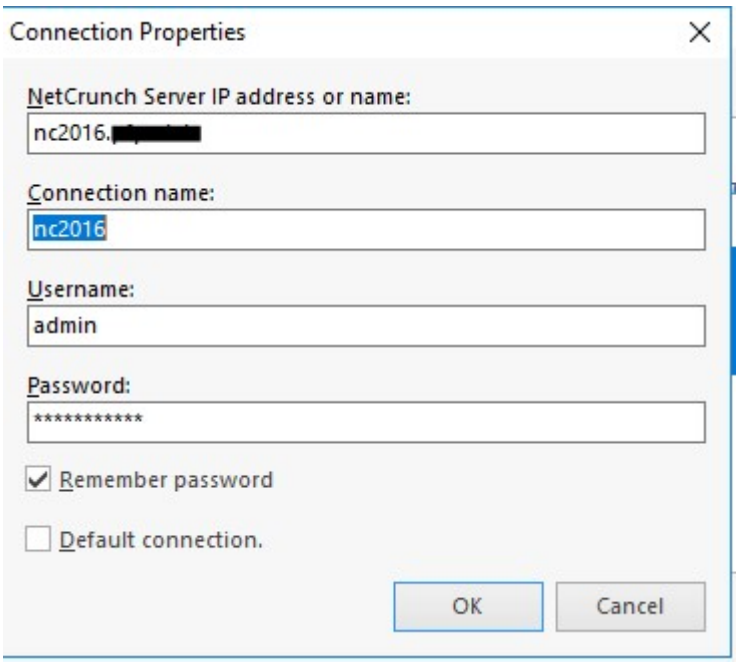

Figure 19 - Connecting to the Client Access Point

4. Double-click the connection to open the Console.
5. Complete these steps on other locations where the console would be installed.
6. Once the console is open and you are able to browse through the Atlas, try failing over. In Failover Cluster Manager, right-click the NetCrunch role and select Move -> Select Node. Select the other server in your cluster and press OK. All NetCrunch services will stop on the server and will start on the other server you just selected.
7. Wait approximately 1 minute as services stop and restart, and the console reconnects to the server. (Tests conducted at the writing of this document averaged 30 seconds for all services to stop then start, and an addition 18 seconds for the console to fully reconnect.)
8. Add more nodes in to monitor. Allow a few minutes to pass, then try failing back over.
9. Repeat this process a few times till you are comfortable, checking the Event Logs to ensure no error messages are being logged.

More in-depth testing will be required before a full production deployment. Such testing should include disconnecting one NetCrunch server from the network (to simulate hard failures), testing with large numbers of nodes/sensors being monitored, and more.

Conclusion

After trial and error, NetCrunch 9.3.3 can indeed be made highly available utilizing Microsoft Failover Clustering. With failover times averaging 1 minute or less in small test environments, resiliency has now become a part of a NetCrunch installation. The use of MSFC now allows for features such as Cluster Aware Updating (Microsoft's Windows Updating tool for Failover Clusters), as well as the ability to physically separate NetCrunch instances (where possible) to provide for additional resiliency in case of environmental incident (e.g. fire, water, or power event). This additional resiliency and availability can

help ensure an administrator does not miss a notification for a device or service being down or degraded, saving the company money in the long run.