

# SQL Code Name “DENALI” 의 HADR 설치 및 구성 가이드

[Type the document subtitle]

3/3/2011

Microsoft Corporation

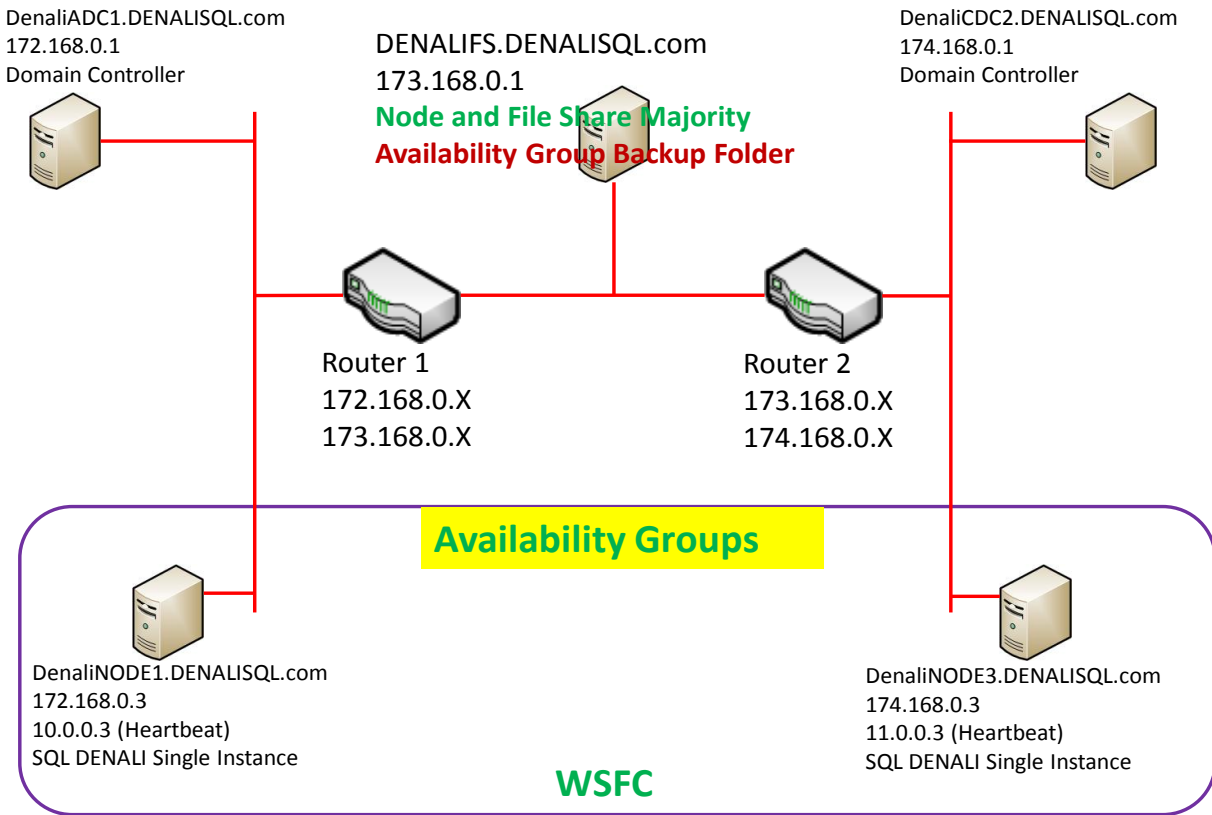
이 동 철 부장

## Contents

데모 환경.....	2
HADR 을 위한 Multi-Subnet Failover Clustering 설치 및 구성 .....	3
Failover Clustering 기능 설치.....	3
Failover Clustering 노드의 Heartbeat 네트워크 구성.....	4
HADR 을 위한 Multi-Subnet Failover Clustering 구성 검증 및 설치 .....	7
Multi-Subnet Failover Clustering 환경에서의 Heartbeat 및 DNS 설정 구성.....	18
Heartbeat 설정 및 구성 .....	18
DNS 설정 및 구성 .....	19
SQL Code Name “Denali” HADR 용 첫 번째 노드 설치하기 .....	24
SQL Code Name “Denali” HADR 용 두 번째 노드 설치하기 .....	28
SQL Code Name “Denali” 에서 HADR 기능 구현 .....	30
HADR 개요 .....	30
Availability Groups .....	30
Multiple Secondaries .....	32
Readable Secondaries .....	33
HADR 구성에 참여할 노드의 SQL 인스턴스의 “HADR” 활성화.....	35
HADR 기능 테스트를 위한 Availability Group 생성.....	36
HADR 의 Availability Group 의 데이터 동기화 작업 .....	41
Primary Replica 에 연결하기 위한 Virtual Network Name 을 Failover Clustering 리소스 생성.....	44
테스트 .....	49
HADR 에 대한 DENALI CTP 1 의 Known Issue .....	51
참조 자료.....	52

## 데모 환경

아래와 같이 데모 환경을 구성해 보았습니다.



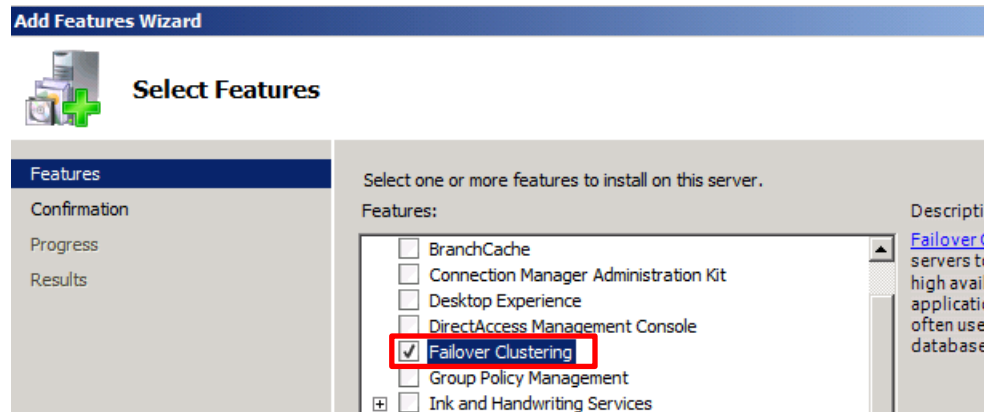
실제 Disaster Recovery 환경과 유사하게 2 개의 노드가 별도의 다른 네트워크 상에 존재하도록 구성해 보았습니다. 전체 서버들의 OS 는 Windows Server 2008 R2 영문으로 구성하였습니다.

## HADR 을 위한 Multi-Subnet Failover Clustering 설치 및 구성

SQL Code Name "Denali" 의 HADR(High Availability Disaster Recovery) 기능 구현을 위해서는 Failover Clustering 이 필수적으로 구성되어야 합니다. 이번 테스트 환경에서는 Multi-Subnet 네트워크 환경 내에서 각기 다른 subnet 에 존재하는 2 노드를 Failover Clustering 의 노드로 구성합니다. 이렇게 구성된 Multi-Subnet Failover Clustering 환경에서 각기 노드에 Denali 를 로컬 인스턴스로 설치한 후에, HADR 기능을 활성화하여, HADR 기능을 구현할 예정입니다.

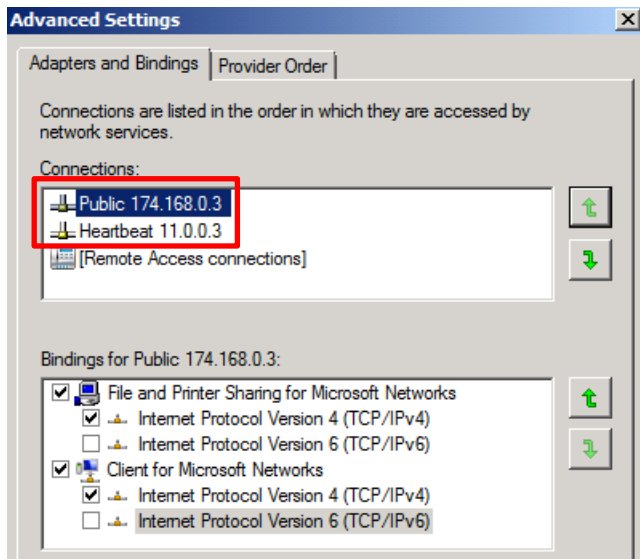
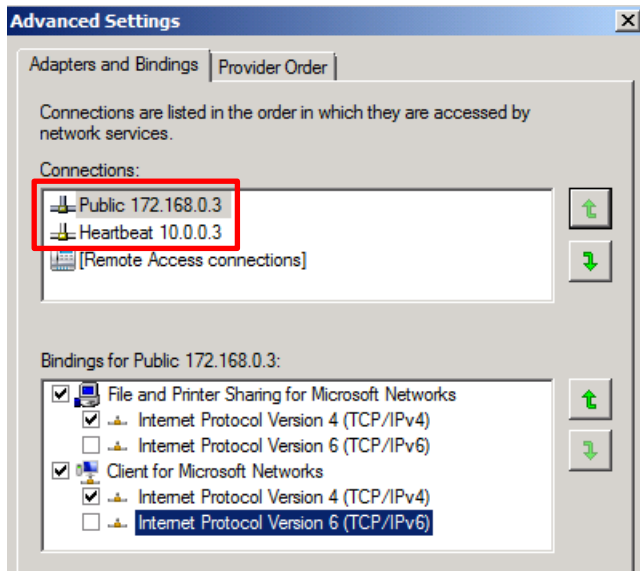
### Failover Clustering 기능 설치

Multi-Subnet Failover Clustering 에 참여할 노드 각각에 아래와 같이 "Failover Clustering" 기능을 설치합니다.



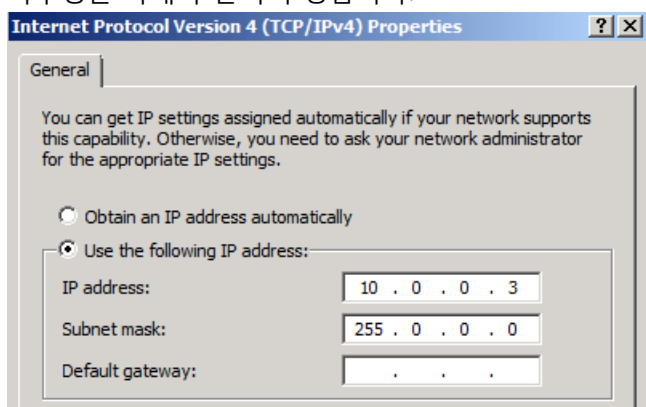
## Failover Clustering 노드의 Heartbeat 네트워크 구성

Multi-subnet Failover Clustering 에 참여할 노드 각각에 Network Adapter 가 2 개 이상이 설치되어 있음을 확인하고, 아래와 같이 바인딩 순서를 지정합니다. 아래와 같이 공용 네트워크를 heartbeat 네트워크 보다 바인딩 순서를 상위로 지정하는 것은 MSCS 및 Failover Clustering 구축 시에 기본입니다. 본 테스트 환경에서는 Failover Clustering 에 참여하는 2 노드가 각기 다른 subnet 에 존재하므로, heartbeat 네트워크도 다른 subnet 임을 주의해야 합니다.



위의 Heartbeat 네트워크 카드의 기본 게이트웨이 설정은 하지 않고, 아래와 같이 정적 라우팅을 설정합니다. 이번 테스트 환경에서는 Failover Clustering 에 참여할 노드들이 다른 서브넷에 존재하므로,

Heartbeat 네트워크도 다른 서브넷에 존재합니다. 그러므로, 이러한 Heartbeat 네트워크에 대한 정적 라우팅을 아래와 같이 구성합니다.



아래와 같은 명령어를 수행하여, 상대방 노드의 Heartbeat 네트워크에 대한 라우팅 경로를 설정합니다.

```
C:\Users\administrator.DENALISQL>route add -p 11.0.0.0 mask 255.0.0.0 10.0.0.254
OK!
```

Route Print 명령어를 수행하여, 위에서 설정한 정적 라우팅이 정상적으로 추가되었는지 확인합니다.

```
Persistent Routes:
=====
Network Address          Netmask  Gateway Address  Metric
-----
0.0.0.0                  0.0.0.0  172.168.0.254   Default
11.0.0.0                 255.0.0.0  10.0.0.254      1
=====
```

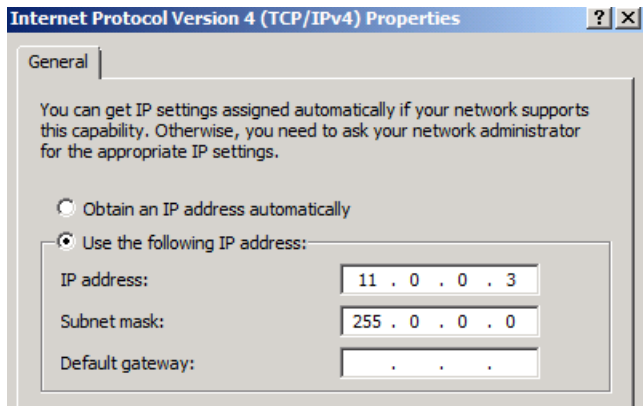
상대방 노드의 Heartbeat 네트워크의 IP 주소에 대한 Ping 테스트를 수행하여, 연결이 정상임을 확인합니다.

```
C:\Users\administrator.DENALISQL>ping 11.0.0.3

Pinging 11.0.0.3 with 32 bytes of data:
Reply from 11.0.0.3: bytes=32 time=1ms TTL=127
Reply from 11.0.0.3: bytes=32 time<1ms TTL=127
Reply from 11.0.0.3: bytes=32 time=1ms TTL=127
Reply from 11.0.0.3: bytes=32 time<1ms TTL=127

Ping statistics for 11.0.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

위에서 수행했던 작업을 Failover Clustering 의 다른 쪽 노드에 대해서도 동일하게 수행합니다.



```
C:\Users\administrator.DENALISQL>route add -p 10.0.0.0 mask 255.0.0.0 11.0.0.254
OK!
```

```
=====  
Persistent Routes:  
  Network Address          Netmask  Gateway Address  Metric  
      0.0.0.0             0.0.0.0   174.168.0.254   Default  
     10.0.0.0           255.0.0.0    11.0.0.254      1  
=====
```

```
C:\Users\administrator.DENALISQL>ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data:
Reply from 10.0.0.3: bytes=32 time=1ms TTL=127
Reply from 10.0.0.3: bytes=32 time<1ms TTL=127
Reply from 10.0.0.3: bytes=32 time<1ms TTL=127
Reply from 10.0.0.3: bytes=32 time<1ms TTL=127

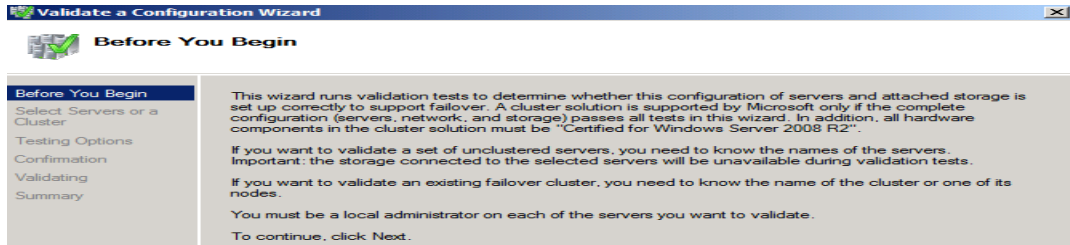
Ping statistics for 10.0.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

이상과 같이 Failover Clustering 의 heartbeat 네트워크 구성을 완료합니다.

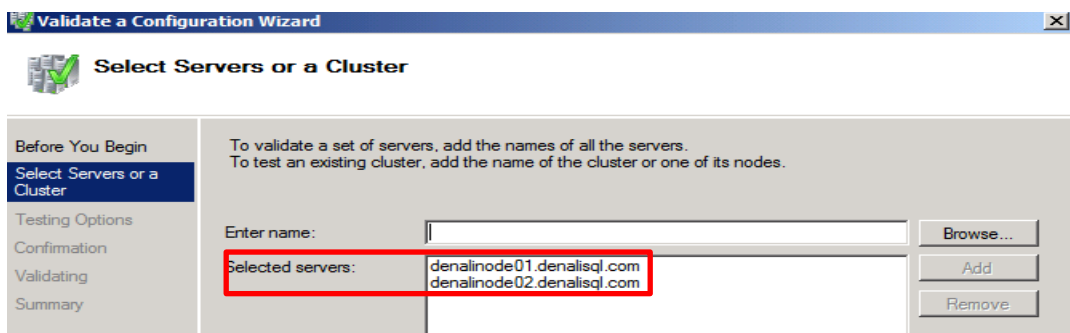
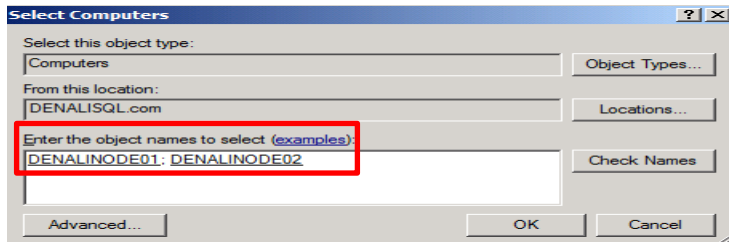
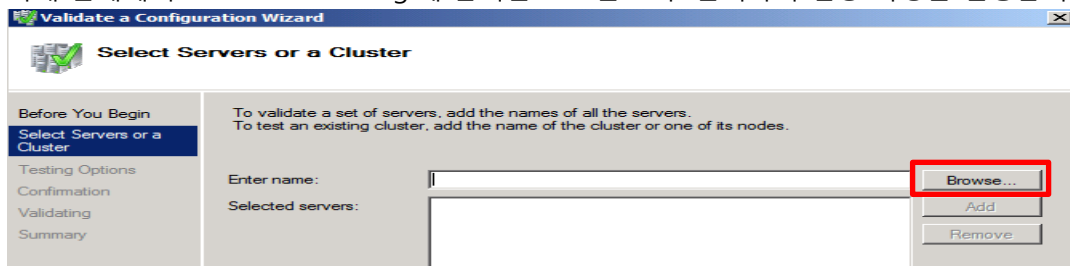
## HADR 을 위한 Multi-Subnet Failover Clustering 구성 검증 및 설치

Multi-Subnet Failover Clustering 을 설치하기 전에, 먼저 "Validate a Configuration" 과정을 수행하여, 구성 검증을 완료해야 합니다. 아래 과정을 수행하여, 검증 과정을 진행합니다.

 Validate a Configuration...

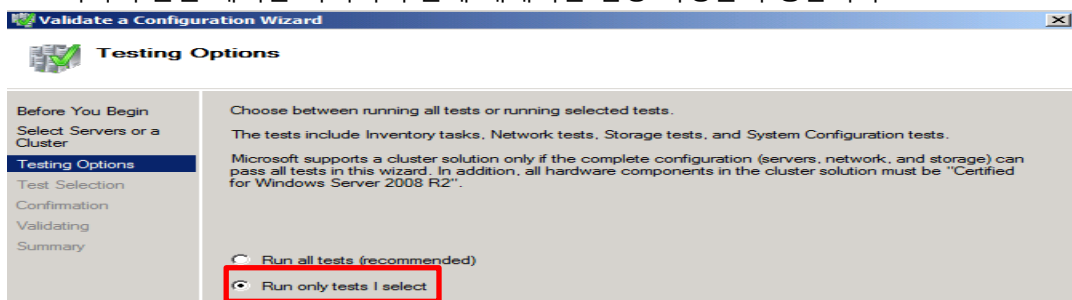


아래 단계에서 Failover Clustering 에 참여할 노드를 모두 선택하여 검증 과정을 진행합니다.

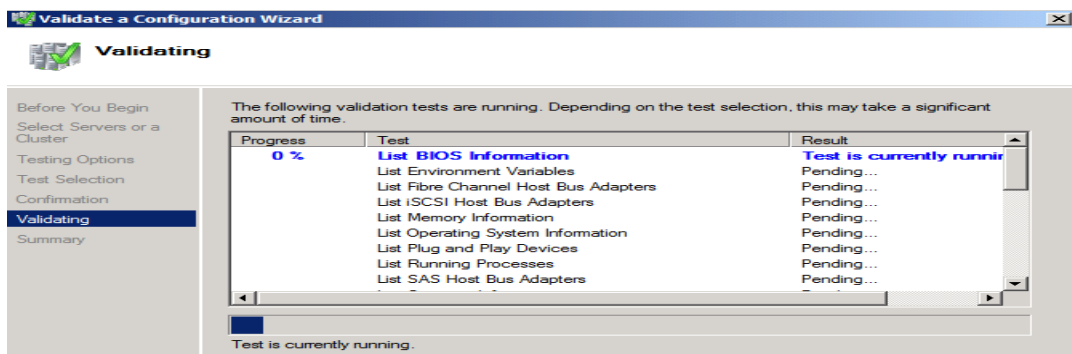
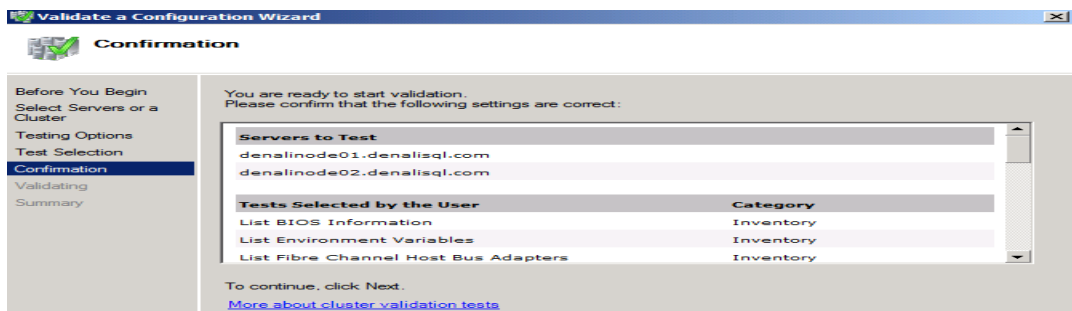
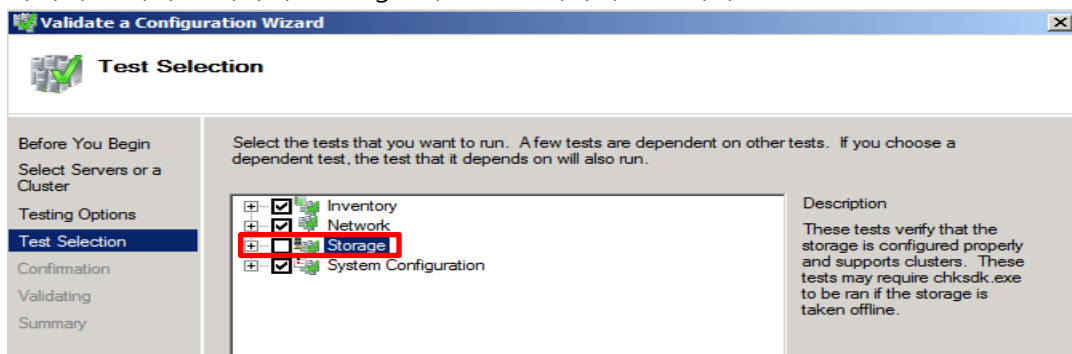




Multi-Subnet Failover Clustering 환경에서는 스토리지 검증 과정이 필요하지 않으므로, 아래 과정에서 검증 요소를 선택해야 합니다. 즉, Multi-Subnet Failover Clustering 환경에서는 별도의 공유 스토리지 없이 제 3의 서버에 Quorum 을 저장하는 “Node and File Share Majority” Quorum 방식을 사용합니다. 스토리지 부분을 제외한 나머지 부분에 대해서만 검증 과정을 수행합니다.

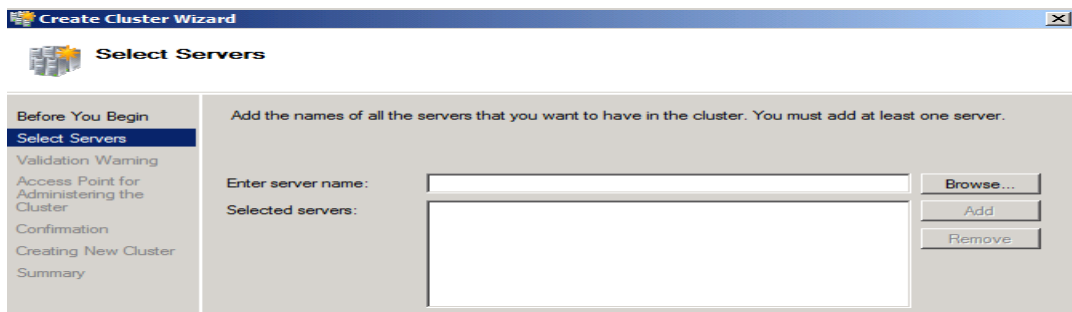
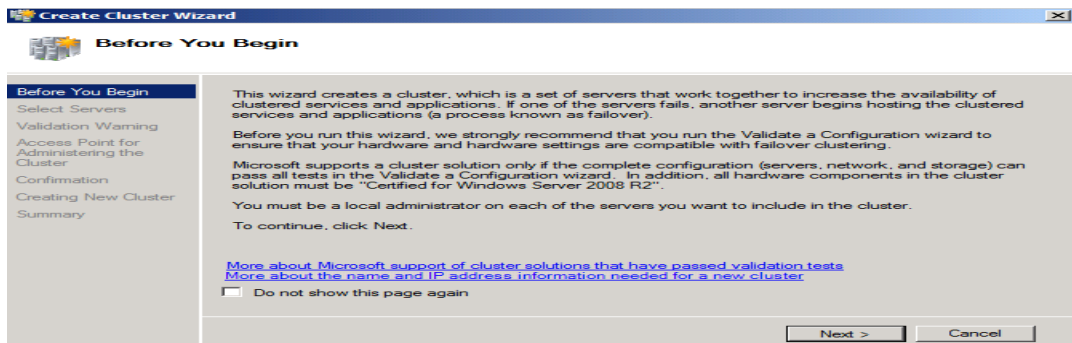


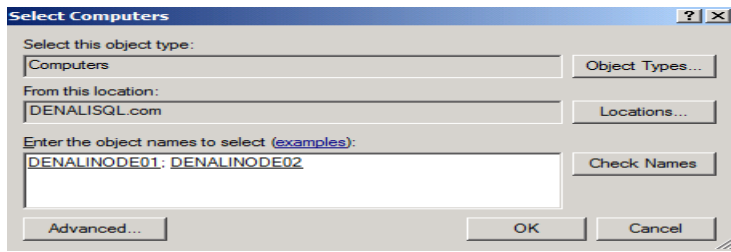
아래 부분에서 명백하게 “Storage” 부분을 선택하지 않습니다.



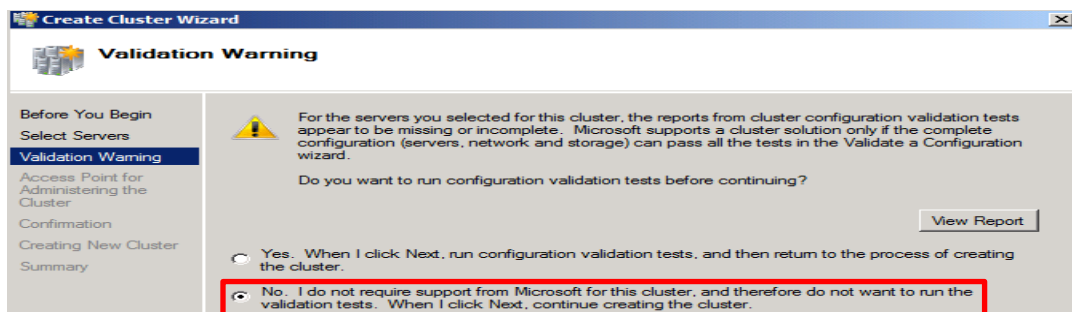
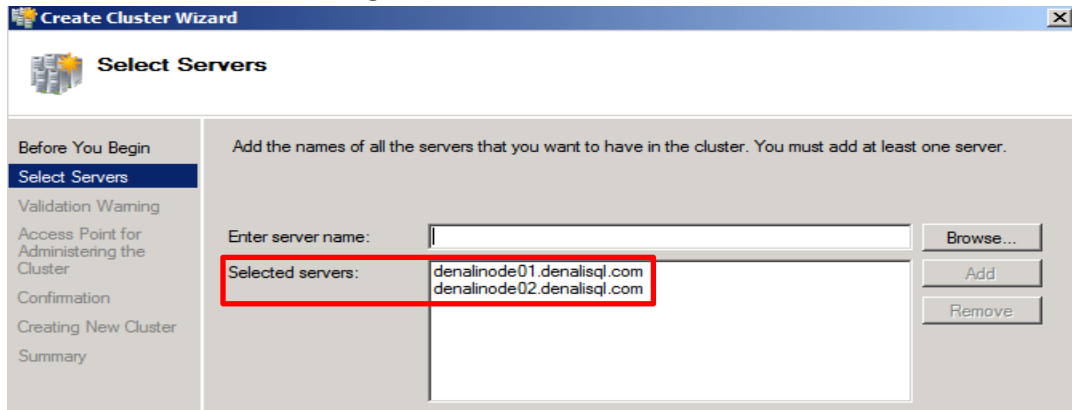


위와 같이 구성 검증 과정이 완료되었으므로, 이제 Multi-Subnet Failover Clustering 설치를 진행합니다.





아래와 같이 Failover Clustering 에 참여할 노드가 모두 선택되었음을 확인하고 다음을 진행합니다.

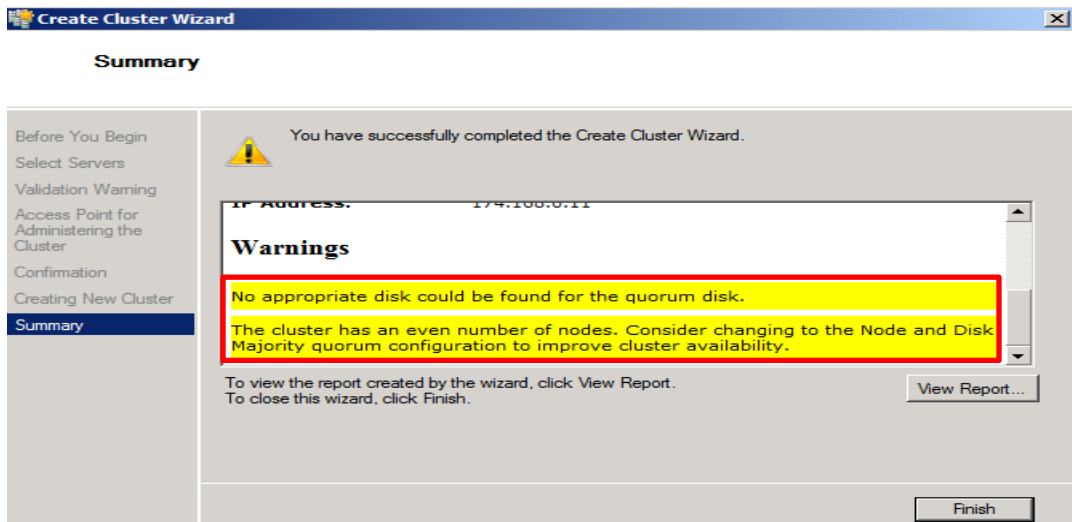
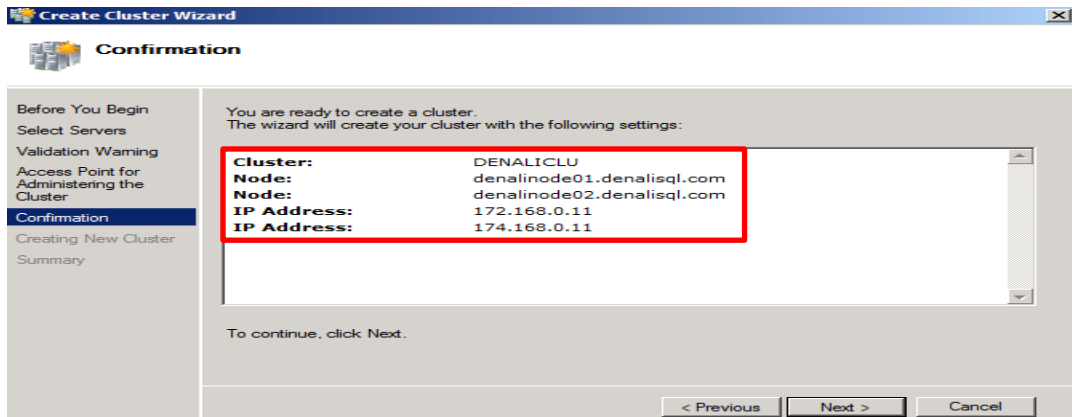
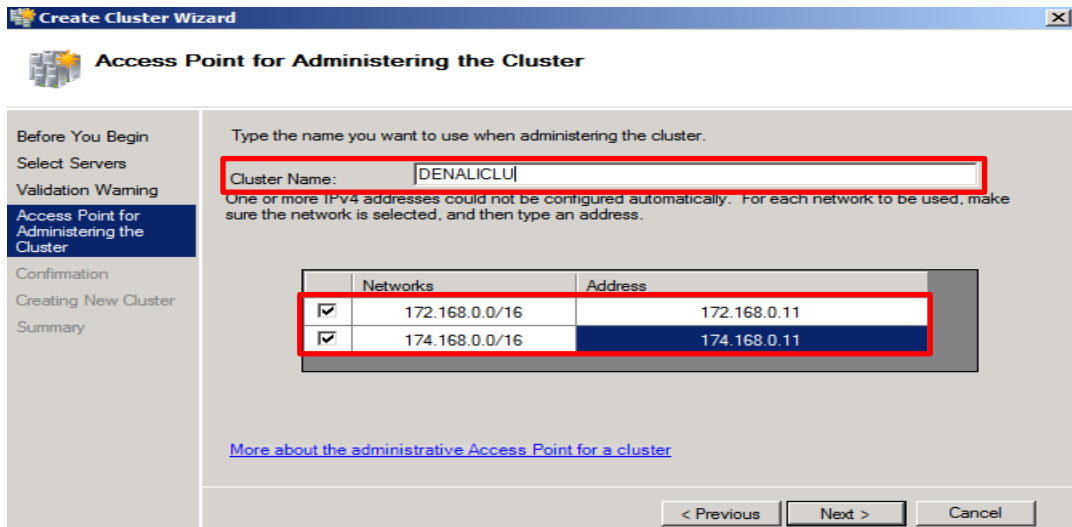


동일 Subnet Failover Clustering 이 아닌 Multi-Subnet Failover Clustering 환경에서는 아래와 같이 2 노드가 참여하고 있는 각기 다른 네트워크 대역에 대해서 각기 클러스터 그룹 관리 IP 를 설정해야 합니다.

- 172.168.0.x (DENALINODE01.DENALISQL.com 서버가 참여한 네트워크 대역)
- 174.168.0.x (DENALINODE02.DENALISQL.com 서버가 참여한 네트워크 대역)

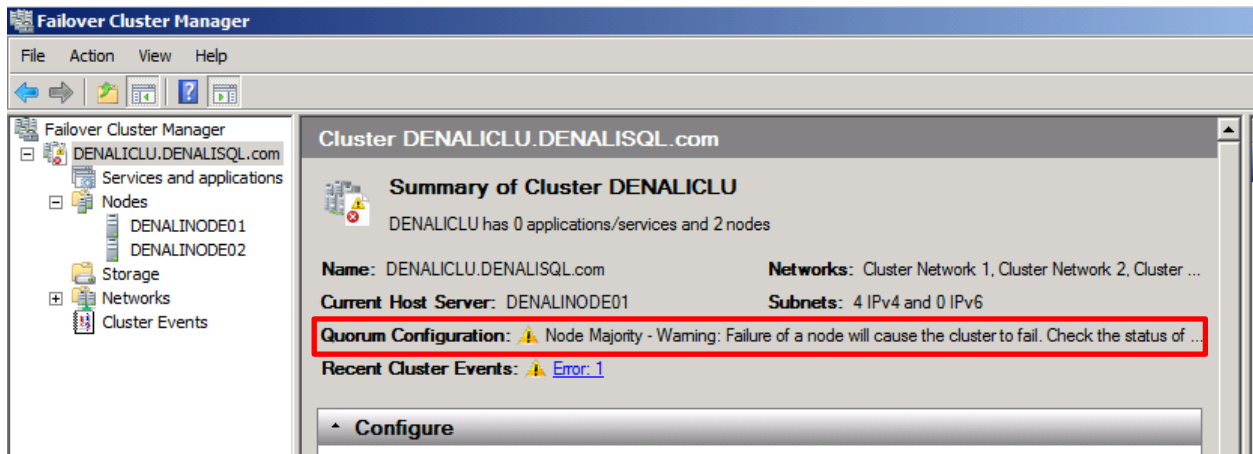
아래 단계에서 위 IP 에 해당하는 Failover Clustering 의 Network Name 도 설정합니다.

- Failover Clustering Network Name : DENALICLU.DENALISQL.com

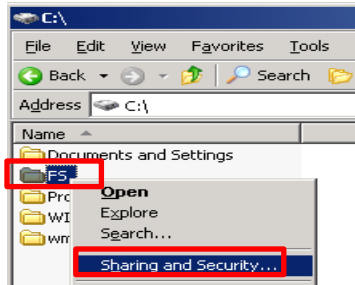


아래와 같이 Multi-Subnet Failover Clustering 설치가 완료되었습니다. 그러나, 위의 설치 요약 보고서를 확인해 보면, Quorum 모드가 "Node Majority" 로 설정되어 있음을 알 수 있습니다. "Node Majority"

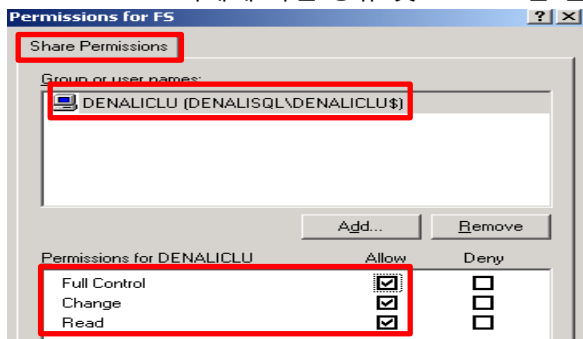
모드에서 2 노드로 구성된 클러스터링 환경에서는 한 쪽 노드가 오프라인이 되면, 전체 클러스터가 오프라인이 됩니다.

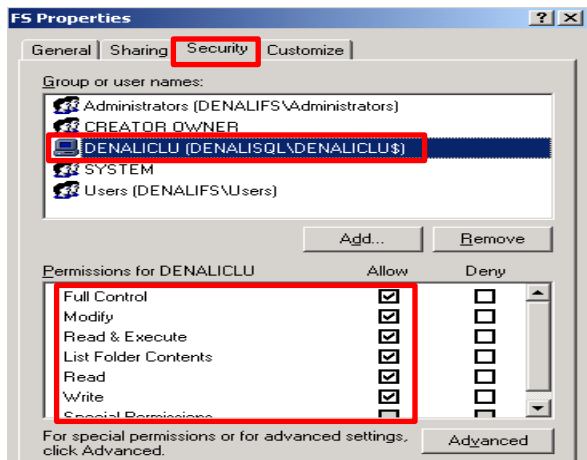


Multi-Subnet Failover Clustering 환경에서의 최적의 Quorum 모드는 "Node and File Share Majority" 입니다. 이 모드로 변경하기 위해 추가적인 "파일 공유 목격자(File Share Witness)" 서버를 구성해야 합니다. 이 서버의 최적 위치는 클러스터링에 참여한 2 노드의 네트워크와는 다른 제 3 의 네트워크이다. 이번 테스트 환경에서는 172.168.0.x 및 174.168.0.x 와는 다른 173.168.0.x 네트워크에 도메인에 조인된 Windows 2003 서버를 이용합니다. 아래와 같이 Witness 로그가 위치할 공유 폴더 구성 및 권한 설정을 합니다.

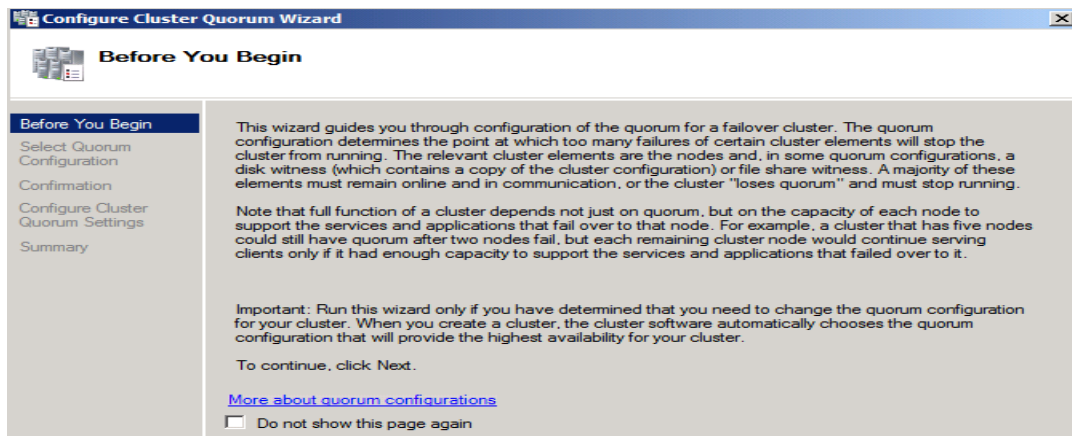
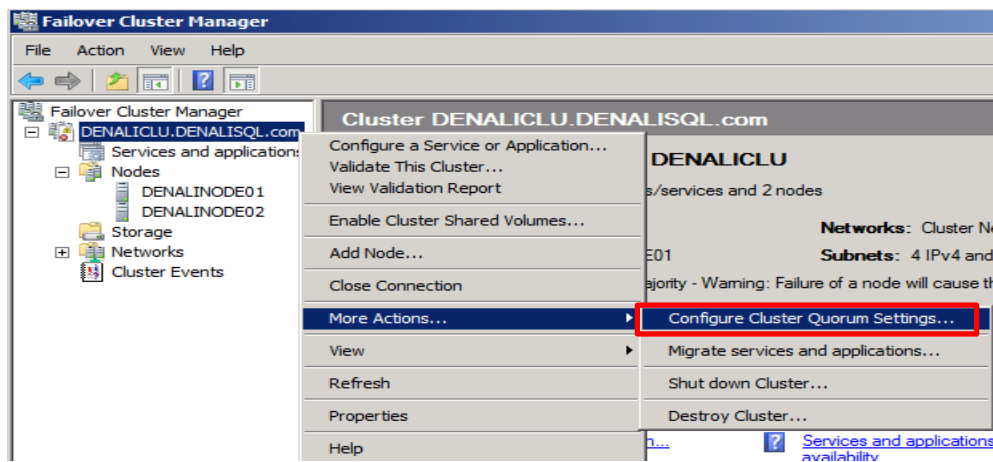


앞서 Multi-Subnet Failover Clustering 의 구성시 생성되었던 클러스터 그룹 네트워크 이름인 "DENALICLU" 객체에 파일 공유 및 NTFS 모든 권한을 설정합니다.

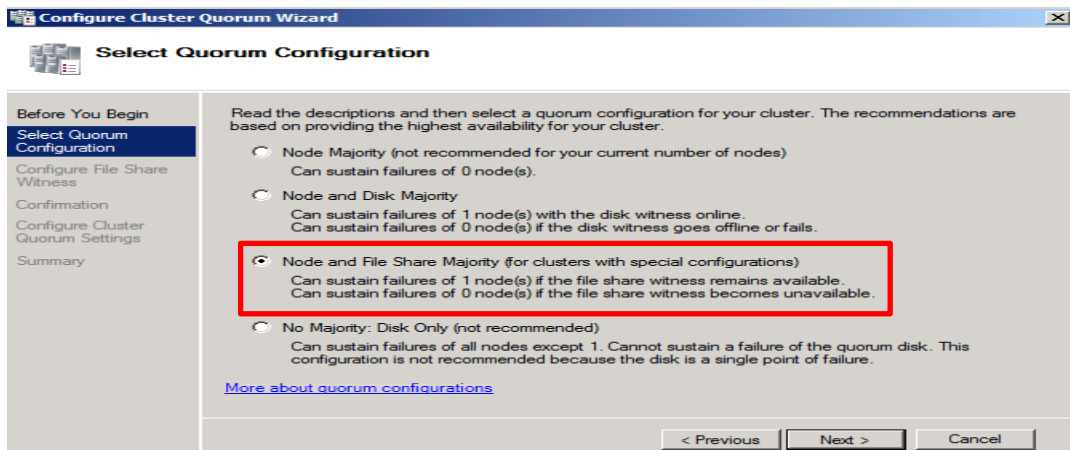




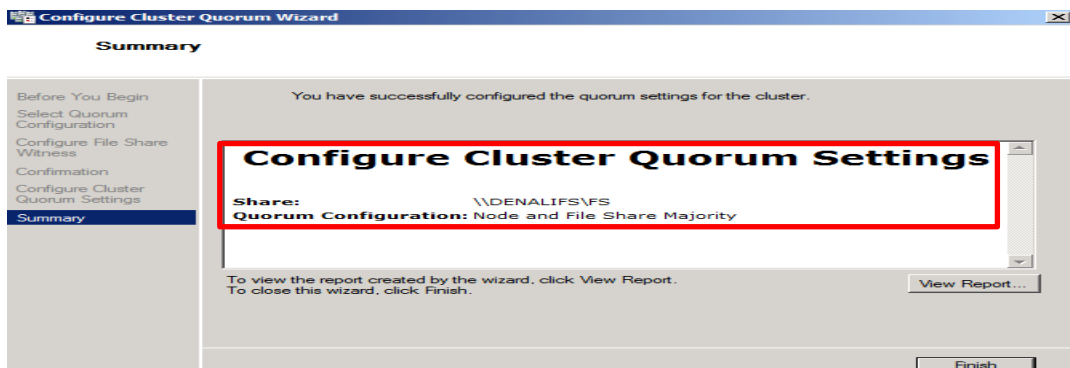
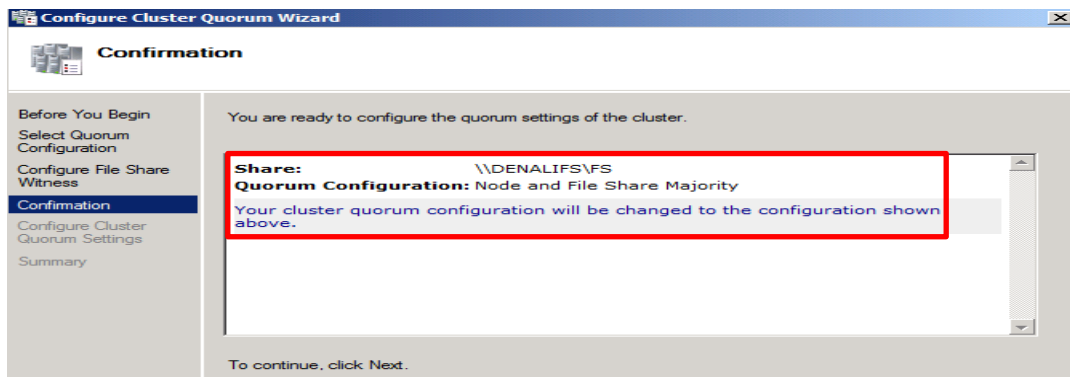
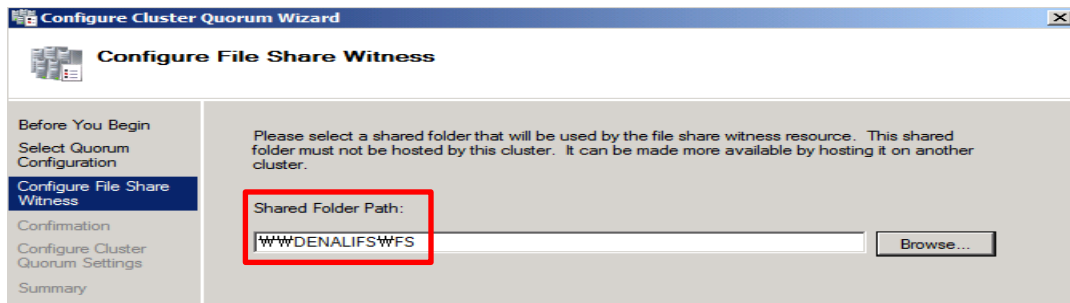
이제 Quorum 모드를 변경합니다.

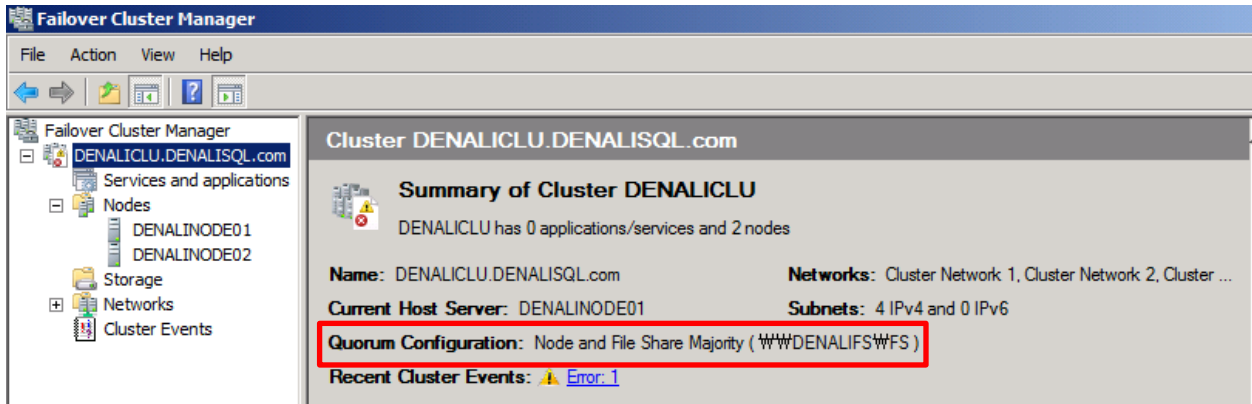


아래 단계에서, "Node and File Share Majority" 방식을 선택합니다.

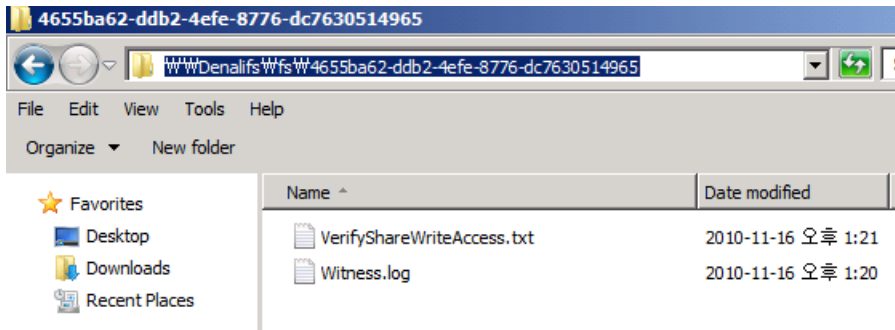


아래 단계에서 Quorum 데이터가 저장될 앞서 지정한 파일 서버의 공유 폴더를 지정합니다.

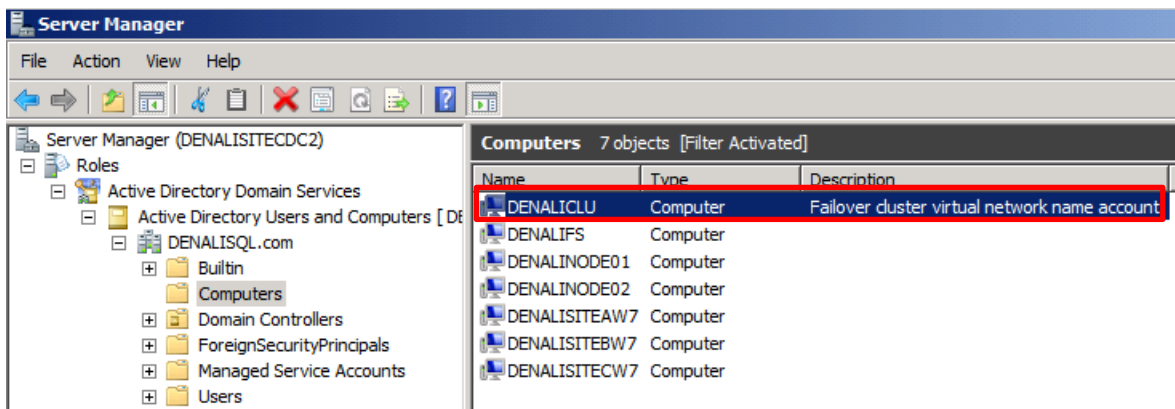




위와 같이 Quorum 모드의 변경이 성공적으로 완료된 후, 아래 그림과 같이 Witness 서버의 공유 폴더에는 witness 로그가 생성되어 있음을 알 수 있습니다.



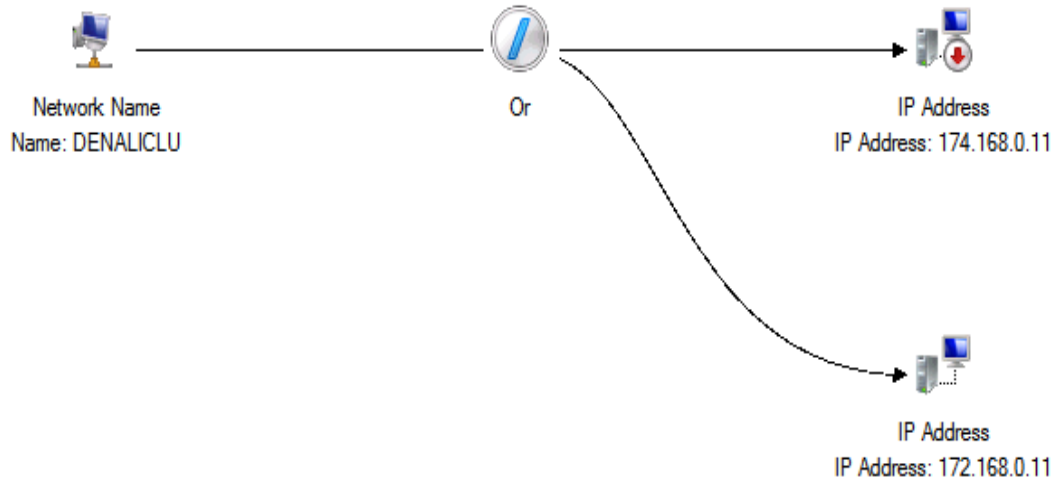
이제 모든 Multi-Subnet Failover Clustering 설치 및 구성이 완료 되었습니다. 아래와 같이 "DENALICLU.DENALISQL.com" 이라는 클러스터 그룹의 네트워크 이름이 DNS 서버 및 ADUC 에 등록되어 있음을 확인할 수 있습니다.





Record Name	Type	Value
(same as parent folder)	Start of Authority (SOA)	[69], denalitecdc2.denalisql.com
(same as parent folder)	Name Server (NS)	denalitecdc2.denalisql.com
(same as parent folder)	Name Server (NS)	denaliteadc1.denalisql.com
(same as parent folder)	Host (A)	172.168.0.1
(same as parent folder)	Host (A)	174.168.0.1
(same as parent folder)	IPv6 Host (AAAA)	2002:aea8:0001:0000:0000
(same as parent folder)	IPv6 Host (AAAA)	2002:aca8:0001:0000:0000
DENALICLU	Host (A)	172.168.0.11
DENALIFS	Host (A)	173.168.0.1

아래 그림은 현재 Cluster Core 자원의 "Dependency Report"를 생성한 결과입니다. "DENALICLU" 라는 네트워크 이름에 대해서 2 개의 IP 자원이 "OR" 로 종속성이 지정되어 있음을 알 수 있습니다.



**Cluster Core Resources**

Name	Status
<b>Cluster Name</b>	
Name: DENALICLU	Online
IP Address: 172.168.0.11	Online
IP Address: 174.168.0.11	Offline
<b>File Share Witness</b>	
File Share Witness (\\DENALIFS\FS)	Online

이와 같이 구성되어 있을 경우에, DENALINODE01 서버가 오프라인 되었을 경우, 클러스터 관리 그룹의 소유권이 DENALINODE02 서버로 이동됩니다. 이때, 클러스터 관리 그룹의 IP 자원은 174.168.0.11 로

변경될 것이고, DENALICLU.DENALISQL.com 이라는 클러스터 관리 그룹의 네트워크 이름 자원은 174.168.0.11 과 바인딩 될 것입니다. 그러나, 여전히 DNS 서버에는 DENALICLU.DENALISQL.com 서버에 대해서 172.168.0.11 이라는 IP 자원이 등록되어 있습니다. 물론, 이 부분은 기본 15 분 정도의 시간이 흐르게 되면 자연스럽게 해결됩니다. 15 분 이라는 시간을 허용할 수 없는 환경이라면, 다음 단계에서 Multi-Subnet Failover Clustering 환경에 맞는 Heartbeat 및 DNS 환경 구성을 해야 합니다.

## Multi-Subnet Failover Clustering 환경에서의 Heartbeat 및 DNS 설정 구성

다중 서브넷을 가진 Multi-Subnet Failover Clustering 환경에서, 클라이언트가 실제 체험하는 다운타임 시간은 장애조치 속도보다는 DNS 복제 속도 및 갱신된 DNS 정보에 대한 질의 속도에 따라 좌우되는 경우가 많습니다. 장애조치 속도, DNS 복제 속도 및 갱신된 DNS 정보에 대한 질의 속도를 최적화하기 위해서는 클러스터의 Heartbeat 네트워크 설정 및 DNS 설정을 최적으로 구성해야 합니다.

### Heartbeat 설정 및 구성

서브넷 구성에 무관하게, "Heartbeat의 주기(Subnet Delay)"는 매 1 초(1000 milliseconds)이다. 아래 명령어를 통해 기본적으로 설정된 "Heartbeat의 주기(Subnet Delay)"를 확인할 수 있습니다.

```
cluster /cluster:<ClusterName> /prop
```

```
PS C:\Users\administrator.DENALISQL> cluster /cluster:DENALICLU /prop
```

```
D DENALICLU CrossSubnetDelay 1000 (0x3e8)
D DENALICLU CrossSubnetThreshold 5 (0x5)
```

```
D DENALICLU SameSubnetDelay 1000 (0x3e8)
D DENALICLU SameSubnetThreshold 5 (0x5)
```

위 결과 값을 보면, "CrossSubnetDelay" 와 "SameSubnetDelay" 2 가지 종류의 "Heartbeat의 주기(Subnet Delay)" 가 있음을 알 수 있습니다.

- CrossSubnetDelay : 다른 서브넷에 존재하는 Failover Clustering 환경에서의 "Heartbeat의 주기(Subnet Delay)". 이 값은 250 ~ 4000 milliseconds 범위내에서 변경할 수 있음.
- SameSubnetDelay : 동일 서브넷에 존재하는 Failover Clustering 환경에서의 "Heartbeat의 주기(Subnet Delay)". 이 값은 250 ~ 2000 milliseconds 범위내에서 변경할 수 있음.

그 외 "SubnetThreshold" 라는 속성 값이 있는데, 이 값은 기본적으로 "5"로 설정되어 있음을 알 수 있습니다. 즉, 기본적으로 5 번의 heartbeat 에서 노드가 반응을 보이지 않는다면, 다른 노드로 장애조치가 발생할 것입니다.

- CrossSubnetThreshold : 다른 서브넷에 존재하는 Failover Clustering 환경에서의 heartbeat 시도하는 횟수. 이 값은 3 번 ~ 10 번 범위내에서 변경할 수 있음.
- SameSubnetThreshold : 동일 서브넷에 존재하는 Failover Clustering 환경에서의 heartbeat 시도하는 횟수. 이 값은 3 번 ~ 10 번 범위내에서 변경할 수 있음.

이러한 값들을 조절함으로써 좀 더 빠른 장애조치를 일으킬 수 있습니다. 이 값들을 변경하기 위해서는 아래와 같은 명령어를 사용합니다.

```
cluster /cluster:<ClusterName> /prop SameSubnetDelay=<value>  
cluster /cluster:<ClusterName> /prop SameSubnetThreshold=<value>  
cluster /cluster:<ClusterName> /prop CrossSubnetDelay=<value>  
cluster /cluster:<ClusterName> /prop CrossSubnetThreshold=<value>
```

또한, 현재 구성되어 있는 네트워크 이름 자원을 아래와 같은 명령어로 확인 할 수 있습니다.

```
cluster /cluster:<ClusterName> /res
```

```
PS C:\Users\administrator.DENALISQL> cluster /cluster:DENALICLU res  
Listing status for all available resources:
```

Resource	Group	Node	Status
Cluster IP Address	Cluster Group	DENALINODE01	Online
Cluster IP Address	174.168.0.11 Cluster Group	DENALINODE01	Offline
Cluster Name	Cluster Group	DENALINODE01	Online
File Share Witness	Cluster Group	DENALINODE01	Online

## DNS 설정 및 구성

먼저, 아래 명령어를 통하여, 기본적으로 설정되어 있는 DNS 관련 항목들을 확인합니다.

```
Get-ClusterResource "Cluster Name" | Get-ClusterParameter
```

```
PS C:\Users\administrator.DENALISQL> Get-ClusterResource "Cluster Name" | Get-ClusterParameter
```

Object	Name	Value	Type
Cluster Name	Name	DENALICLU	String
Cluster Name	DnsName	DENALICLU	String
Cluster Name	RemapPipeNames	0	UInt32
Cluster Name	HostRecordTTL	1200	UInt32
Cluster Name	RegisterAllProvidersIP	0	UInt32
Cluster Name	PublishPTRRecords	0	UInt32
Cluster Name	TimerCallbackAdditionalThr...	5	UInt32
Cluster Name	ResourceData	{1, 0, 0, 0...}	ByteArray
Cluster Name	StatusNetBIOS	0	UInt32
Cluster Name	StatusDNS	0	UInt32
Cluster Name	StatusKerberos	0	UInt32
Cluster Name	CreatingDC	\\WDENALISITEADC1.DENALISQL...	String
Cluster Name	LastDNSUpdateTime	2010-11-16 오전 3:57:39	DateTime
Cluster Name	ObjectGUID	969de2badcae344bbfa161e670...	String

위 항목 중에서 Multi-Subnet Failover Clustering 환경에서 장애조치시에, 클라이언트가 가장 빠르게 네트워크 이름 자원을 접근할 수 있도록 하기 위해서 변경 가능한 값들은 아래와 같습니다.

- ①. HostRecordTTL : 클러스터 네트워크 이름의 DNS 서버에 등록 된 후, DNS 서버에 캐쉬되는 TTL(Time To Live) 값입니다. 이 값은 기본적으로 1200 초(20 분)로 설정되어 있다. 즉, 이 문서의 테스트 환경을 기준으로 설명하게 되면, 초기에 DENALICLU.DENALISQL.com 이라는 클러스터 네트워크 이름이 DNS 서버에 172.168.0.11 로 등록됩니다. 이 값은 기본적으로 20 분 동안, DNS 서버의 캐쉬에 저장되어 있습니다. 그러나, 이 클러스터가 다른 서브넷 대역인 174.168.0.x 대역으로 장애조치가 발생하게 되면, 클라이언트들은 20 분 동안 이 클러스터에 접근할 수 없게 됩니다. 왜냐하면, 클라이언트들은 DNS 서버에서 DENALICLU.DENALISQL.com 에 대한 IP 를 이미 오프라인 되어 있는 172.168.0.11 로 받기 때문에, 클라이언트들은 클러스터에 접근할 수 없게 됩니다. 그래서, 이 클러스터가 다른 서브넷 대역인 174.168.0.x 대역으로 장애조치가 발생하게 되면, 또 다른 IP 자원인 174.168.0.11 이 바로 DENALICLU.DENALISQL.com 이라는 클러스터 네트워크 이름으로 DNS 서버에 등록이 되어야만 클라이언트들은 DENALICLU.DENALISQL.com 라는 클러스터에 접근할 수 있습니다. 이 값은 Failover Clustering 에 생성되는 서버 어플리케이션 및 서비스에 따라 달리 조정해야 하는데, 대표적으로 Exchange 2007 & 2010 같은 경우는 300 초(5 분) 가 권장 값입니다. 아래와 같은 명령어로 HostRecordTTL 값을 변경할 수 있습니다.

```
cluster /cluster:<ClusterName> res <NetworkNameResource> /priv
HostRecordTTL=<TimeInSeconds>
```

이번 테스트 환경에서 HostRecordTTL 값을 10 초로 변경합니다.

```
PS C:\Users\Administrator.DENALISQL> cluster /cluster:DENALICLU res "Cluster Name" /priv HostRecordTTL=10

System warning 5024 (0x000013a0).
The properties were stored but not all changes will take effect until the next time the resource is brought online.
```

위와 같이 설정을 변경 한 후, 아래 명령어로 변경된 값을 확인할 수 있습니다.

```
cluster /cluster:<ClusterName> res <NetworkNameResource> /priv
```

```
PS C:\Users\Administrator.DENALISQL> cluster /cluster:DENALICLU res "Cluster Name" /priv
Listing private properties for 'Cluster Name':
```

T	Resource	Name	Value
BR	Cluster Name	ResourceData	01 00 00 00 ... (260 bytes)
DR	Cluster Name	StatusNetBIOS	0 (0x0)
DR	Cluster Name	StatusDNS	0 (0x0)
DR	Cluster Name	StatusKerberos	0 (0x0)
SR	Cluster Name	CreatingDC	WWW.DENALISITEADC1.DENALISQL.com
FTR	Cluster Name	LastDNSUpdateTime	2010-11-16 오후 12:57:39
SR	Cluster Name	ObjectGUID	969de2badcae344bbfa161e670a7db2d
S	Cluster Name	Name	DENALICLU
S	Cluster Name	DnsName	DENALICLU
D	Cluster Name	RemapPipeNames	0 (0x0)
D	Cluster Name	HostRecordTTL	10 (0xa)
D	Cluster Name	RegisterAllProvidersIP	0 (0x0)
D	Cluster Name	PublishPTRRecords	0 (0x0)
D	Cluster Name	TimerCallbackAdditionalThreshold	5 (0x5)

- ②. RegisterAllprovidersIP : 이 값은 기본적으로 "0" 으로 설정되어 있습니다. 즉, Multi-Subnet Failover Clustering 환경에서 네트워크 이름 자원에 대해서 등록되는 IP 자원은 각 서브넷 대역 별로 할당됩니다. 그러나, 여러 개 할당된 IP 자원은 기본적으로 특정 시점에는 당연히 1 개의 IP 자원만 온라인됩니다. 이 값이 "0"이면, 온라인된 하나의 IP 자원만 해당 네트워크 이름 자원과 함께 DNS 등록됩니다. 만약, 이 값을 "1"로 변경한다면, Multi-Subnet Failover Clustering 환경에서 네트워크 이름 자원에 대해서 등록되는 모든 IP 를 DNS 서버에 등록합니다. Multi-Subnet Failover Clustering 환경에서는 이 값을 "1"로 변경하여 모든 IP 자원에 대해서 DNS 서버에 등록되도록 하는 것이 권장 사항입니다.

```
cluster /cluster:<ClusterName> res <NetworkNameResource> /priv RegisterAllProvidersIP=1
```

```
PS C:\Users\Administrator.DENALISQL> cluster /cluster:DENALICLU res "Cluster Name" /priv RegisterAllProvidersIP=1

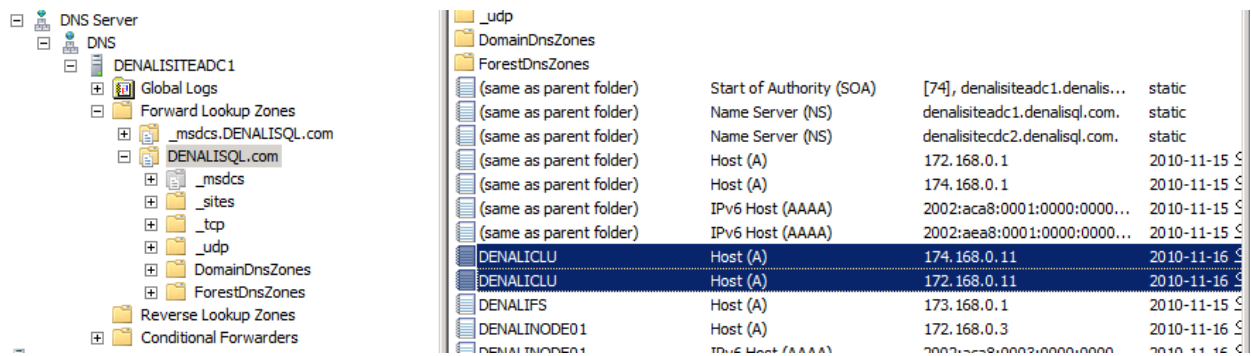
System warning 5024 (0x000013a0).
The properties were stored but not all changes will take effect until the next time the resource is brought online.
```

위 설정된 값을 확인합니다.

```
PS C:\Users\Administrator.DENALISQL> cluster /cluster:DENALICLU res "Cluster Name" /priv
Listing private properties for 'Cluster Name':
```

T	Resource	Name	Value
BR	Cluster Name	ResourceData	01 00 00 00 ... (260 bytes)
DR	Cluster Name	StatusNetBIOS	0 (0x0)
DR	Cluster Name	StatusDNS	0 (0x0)
DR	Cluster Name	StatusKerberos	0 (0x0)
SR	Cluster Name	CreatingDC	WWW.DENALISITEADC1.DENALISQL.COM
FTR	Cluster Name	LastDNSUpdateTime	2010-11-16 오후 12:57:39
SR	Cluster Name	ObjectGUID	969de2badcae344bbfa161e670a7db2d
S	Cluster Name	Name	DENALICLU
S	Cluster Name	DnsName	DENALICLU
D	Cluster Name	RemapPipeNames	0 (0x0)
D	Cluster Name	HostRecordTTL	10 (0xa)
D	Cluster Name	RegisterAllProvidersIP	1 (0x1)
D	Cluster Name	PublishPTRRecords	0 (0x0)
D	Cluster Name	TimerCallbackAdditionalThreshold	5 (0x5)

변경 후, DENALICLU.DENALISQL.com 클러스터 네트워크 이름 자원에 대해서 아래와 같이 2 개의 별도 서브넷에 해당하는 IP 자원이 DNS 서버에 등록되어 있음을 알 수 있습니다.



기타 주의해서 봐야할 속성 값은 아래와 같이 2 가지 정도가 있습니다.

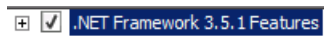
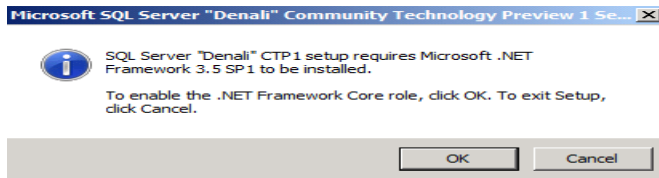
- DnsName : 실제 DNS 서버에 등록되는 클러스터 네트워크 이름임. 이 이름은 총 63 자리 이상을 넘을 수 없습니다.
- PublishPTRRecords : 이 플래그는 클러스터 네트워크 이름의 PTR 레코드를 공개할지 여부를 결정하는 값임. 기본은 공개하지 않는 "0" 값이 설정되어 있음.



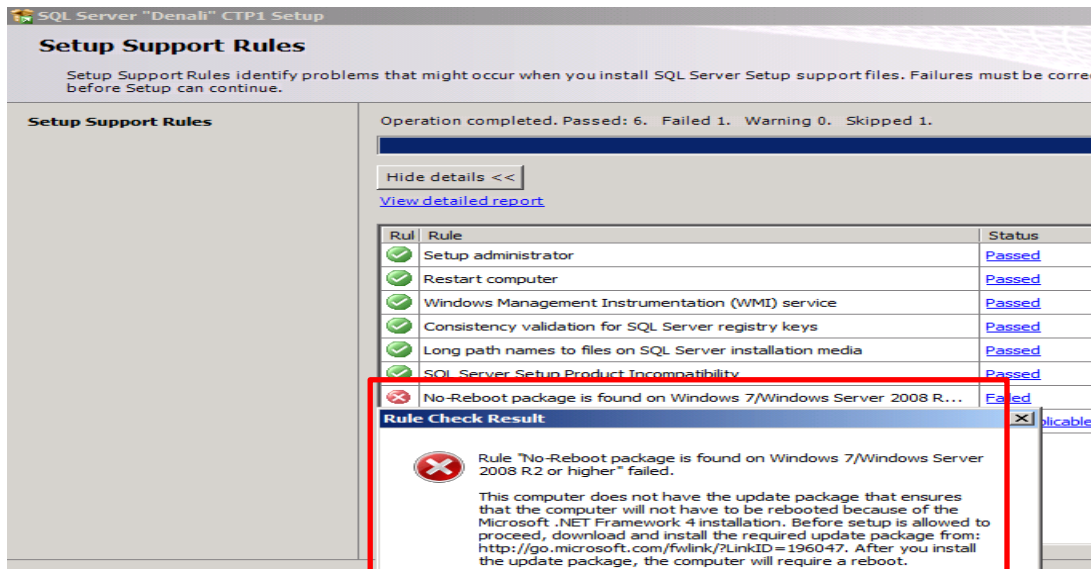
## SQL Code Name “Denali” HADR 용 첫 번째 노드 설치하기

이번 과정에서는 SQL Code Name “Denali” 의 HADR 기능 테스트를 위해, 앞서 Failover Clustering 을 구성했던 2 노드 중에서, 첫 번째 노드에 Single 인스턴스로 Denali 설치를 진행합니다. 여기에서 유념해야 할 사실은 절대 Denali 설치를 SQL 가상 서버로 설치하지 않고, 로컬에 단일 인스턴스로 설치한다는 점입니다. 아래 스크린 샷에 따라 Denali 설치를 완료합니다.

아래와 같이 Denali 설치 전에, 사전 설치 항목으로 “.NET Framework 3.5 SP1”을 Windows Server 2008 R2 의 “기능”에서 추가 설치한다.



아래와 같이, “.NET Framework 3.5 SP1” 의 추가 업데이트 패키지를 설치해야 합니다.



- An update is available for Microsoft .NET Framework 3.5 Service Pack 1 on Windows 7 and Windows Server 2008 R2 (<http://go.microsoft.com/fwlink/?LinkId=196047>)

**SQL Server "Denali" CTP1 Setup**

## Setup Support Rules

Setup Support Rules identify problems that might occur when you install SQL Server Setup support files. Failures must be corrected before Setup can continue.

Operation completed. Passed: 7. Failed 0. Warning 0. Skipped 1.

Hide details << Re-run

[View detailed report](#)

Rul	Rule	Status
✓	Setup administrator	Passed
✓	Restart computer	Passed
✓	Windows Management Instrumentation (WMI) service	Passed
✓	Consistency validation for SQL Server registry keys	Passed
✓	Long path names to files on SQL Server installation media	Passed
✓	SQL Server Setup Product Incompatibility	Passed
✓	No-Reboot package is found on Windows 7/Windows Server 2008 R...	Passed
✓	No-Reboot package is found on Vista/Windows Server 2008	Not applicable

**Install a SQL Server Failover Cluster**

## Setup Support Rules

Setup Support Rules identify problems that might occur when you install SQL Server Setup support files. Failures must be corrected before Setup can continue.

Operation completed. Passed: 20. Failed 0. Warning 3. Skipped 1.

Hide details << Re-run

[View detailed report](#)

Rul	Rule	Status
✓	Fusion Active Template Library (ATL)	Passed
✓	Cluster Node	Passed
✓	Windows Management Instrumentation (WMI) service (DENALINODE...	Passed
✓	Cluster Remote Access (DENALINODE01)	Passed
✓	Cluster service verification	Passed
✓	Distributed Transaction Coordinator (MSDTC) installed (DENALINODE...	Passed
✓	Distributed Transaction Coordinator (MSDTC) service	Passed
⚠	Distributed Transaction Coordinator (MSDTC) clustered	Warning
✓	Microsoft Cluster Service (MSCS) cluster verification errors	Passed
⚠	Microsoft Cluster Service (MSCS) cluster verification warnings	Warning
✓	Remote registry service (DENALINODE01)	Passed
✓	Cluster shared disk available check	Not applicable
✓	Domain controller	Passed

**SQL Server "Denali" CTP1 Setup**

## Product Key

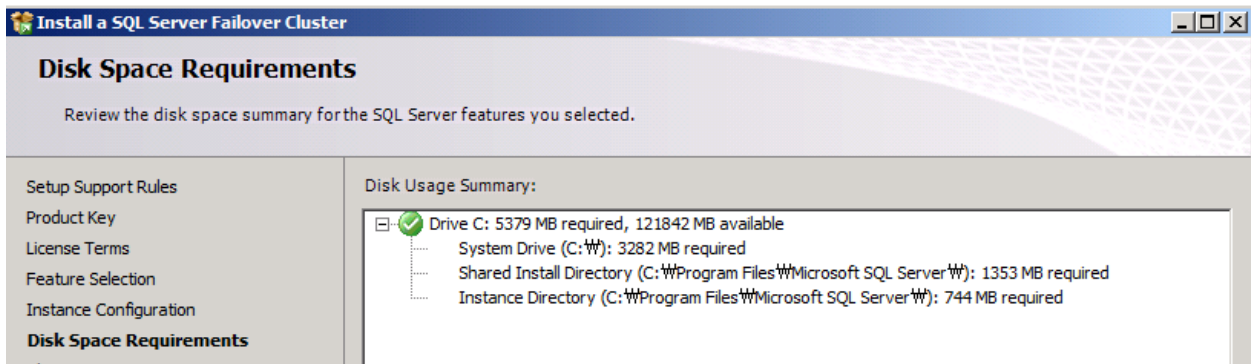
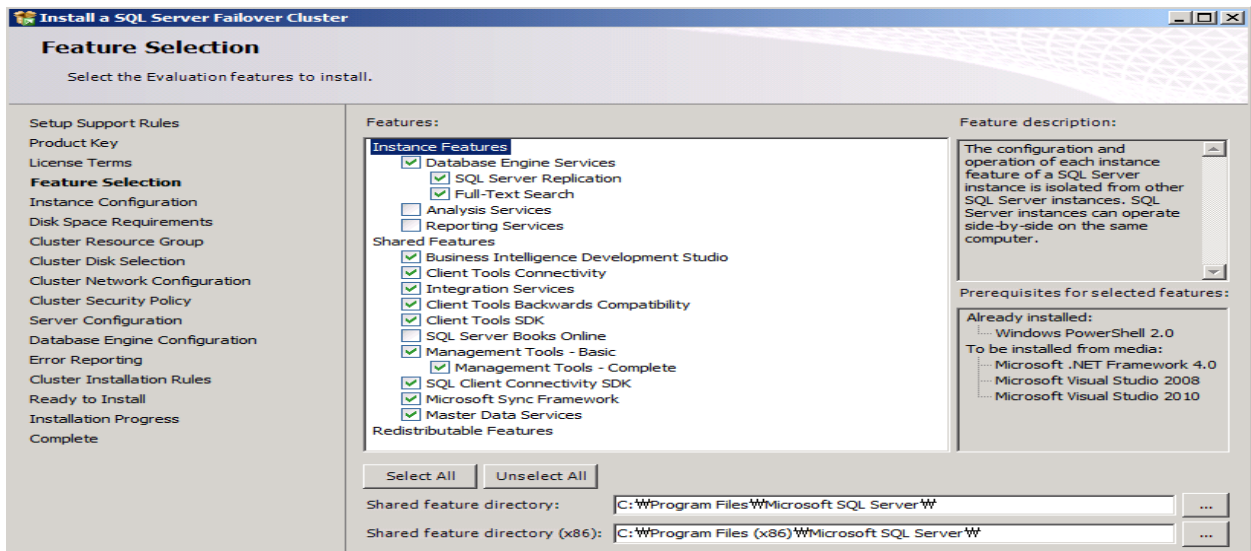
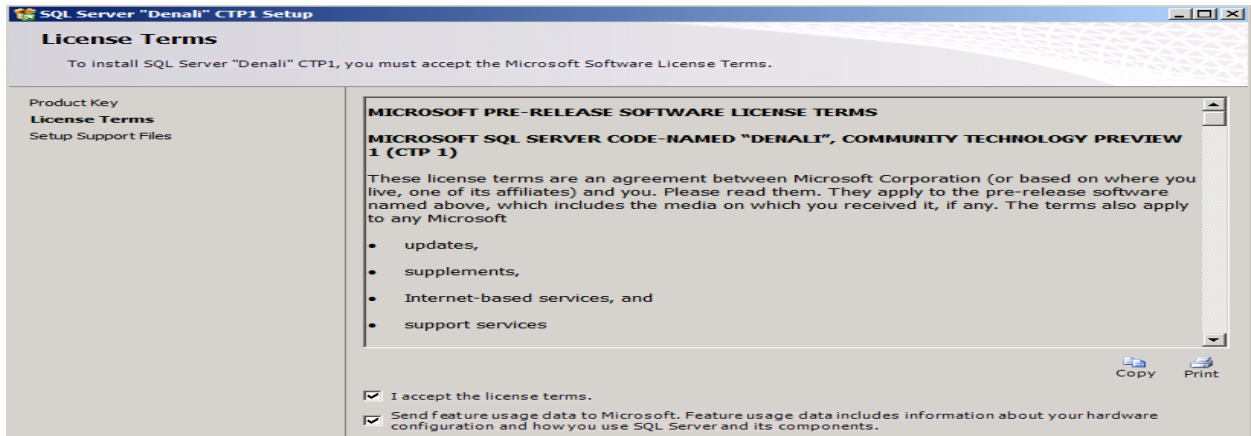
Specify the edition of SQL Server "Denali" CTP1 to install.

Product Key  
License Terms  
Setup Support Files

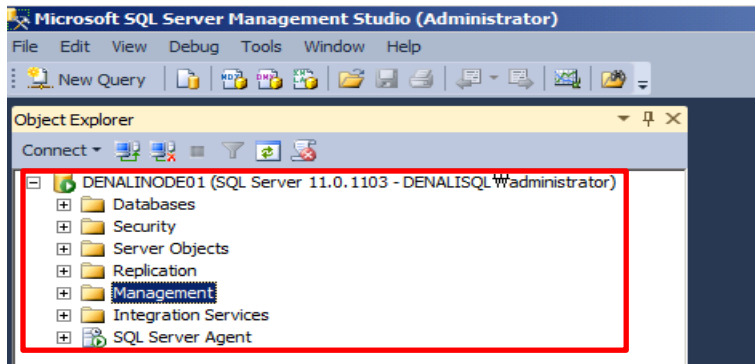
Validate this instance of SQL Server "Denali" CTP1 by entering the 25-character key from the Microsoft certificate of authenticity or product packaging. You can also specify a free edition of SQL Server, such as Evaluation or Express. Evaluation has the largest set of SQL Server features, as documented in SQL Server Books Online, and is activated with a 180-day expiration. To upgrade from one edition to another, run the Edition Upgrade Wizard.

Specify a free edition:

Enter the product key:



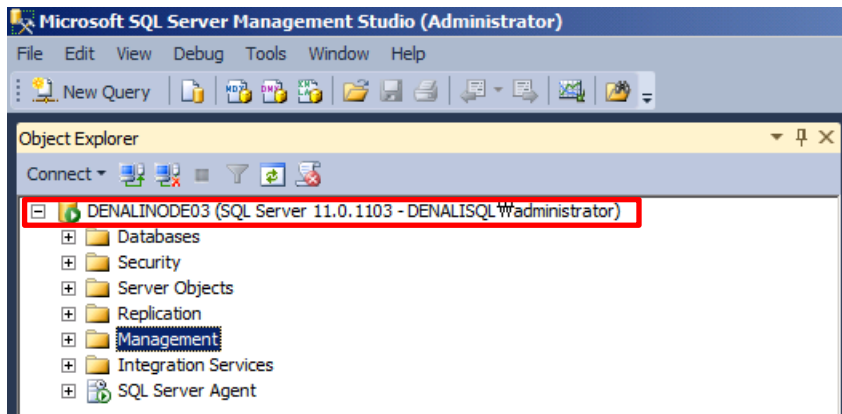
위와 같이 설치 옵션을 설정 한 후, 설치를 완료합니다. 설치 후에 SQL Management Studio 를 이용하여 정상 설치 여부를 확인합니다.



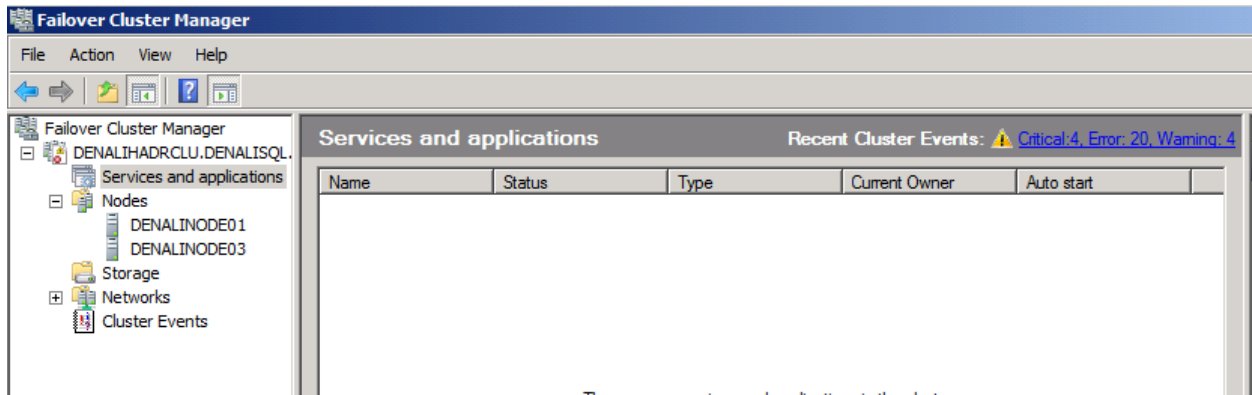
## SQL Code Name “Denali” HADR 용 두 번째 노드 설치하기

앞서 HADR 용 첫 번째 노드에서 Denali 를 로컬 단일 인스턴스로 설치를 완료했습니다. 마찬가지로, Failover Clustering 에 참여한 두 번째 노드에서도 첫 번째 노드와 마찬가지로 Denali 를 로컬 단일 인스턴스로 설치를 완료합니다.

아래는 설치를 완료한 후의 SQL Management Studio 를 통해 정상 설치 여부를 확인한 화면입니다.



지금까지의 작업을 완료 한 후, Failover Clustering 관리자를 확인해 보면, 기본 Failover Clustering 구성 외에 어떠한 서비스나 어플리케이션이 Failover Clustering 이 구성되어 있지 않음을 확인합니다.



혹시 앞서 Multi-subnet Failover Clustering 구성 작업을 유심하게 살펴 보신 분들께서는, 위 그림에서 차이점을 확인하실 수 있을 겁니다. 제가 데모 환경을 구성하다 보니, Cluster Network Name 및 두 번째 노드의 서버 이름을 변경하게 되었습니다. IP 는 변경 사항이 없습니다.

- 두 번째 노드 이름 변경 : DENALINODE02.DENALISQL.com -> **DENALINODE03.DENALISQL.com**
- Cluster Network Name : DENALICLU.DENALISQL.com -> **DENALIHADRCLU.DENALISQL.com**

이렇게 변경된 부분을 고려해서 차후 작업 내용을 읽어주셨으면 합니다.

이제 HADR 기능 구현을 위한 구성 작업을 시작합니다.

## SQL Code Name “Denali” 에서 HADR 기능 구현

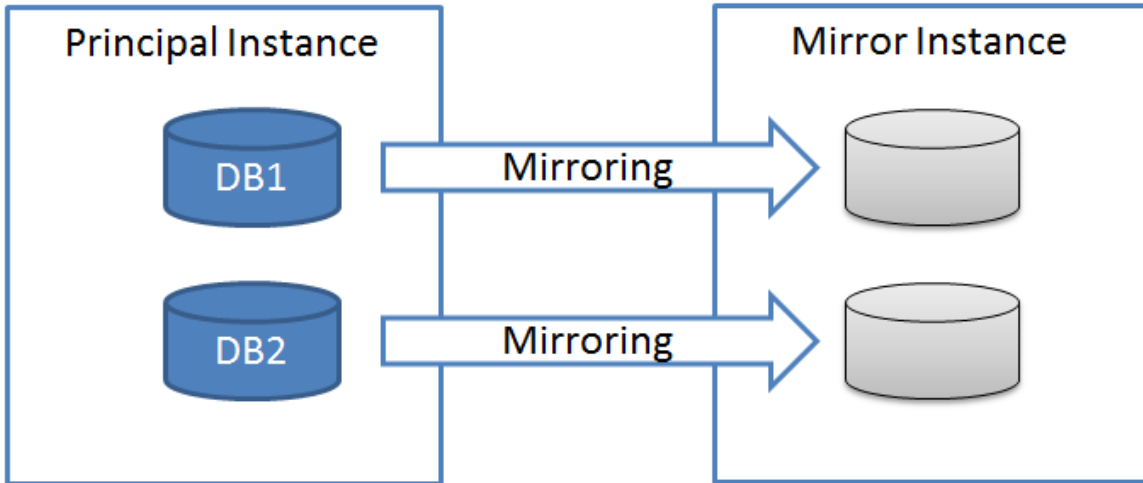
### HADR 개요

HADR 은 SQL Code Name “Denali” 제품에서 소개된 기능입니다. HADR 은 Microsoft 사에서 소개할 때, “AlwaysOn 또는 Hadron” 기술이라고 합니다. HADR 은 Denali 이전의 SQL 서버 제품들이 제공했던, 데이터베이스 미러링 기술을 계승했을 뿐만 아니라, 좀 더 많은 기술적인 진전이 있는 기능입니다. 일단 기본적으로, HADR 은 데이터베이스 미러링과 유사하게 트랜잭션 로그를 전달함으로써 데이터베이스의 물리적인 복제를 기반으로 제공되는 기술입니다. 실제로는, HADR 은 “데이터베이스 미러링(Database Mirroring)”과 유사할 뿐만 아니라, 실제로 데이터베이스를 복제하기 위해 데이터베이스 미러링에서 사용했던 기술을 동일하게 사용합니다. 즉, HADR 을 설정 단계는 “미러링 세션(Mirroring Session)”을 설정하는 단계를 포함하고 있습니다. 또한, HADR 을 모니터링 및 설정 단계에 “미러링 엔드포인트(Mirroring Endpoint)”, “카탈로그 뷰(Catalog View)” 및 DMV 가 실제 사용됩니다. 이런 점을 고려하면, HADR 과 기존 “데이터베이스 미러링(Database Mirroring)”과의 차이점을 확인할 수 없습니다. 그러나, 아래 3 가지 HADR 의 구성 요소로 인해 “데이터베이스 미러링(Database Mirroring)”과는 비교할 수 없는 기능을 제공하게 됩니다.

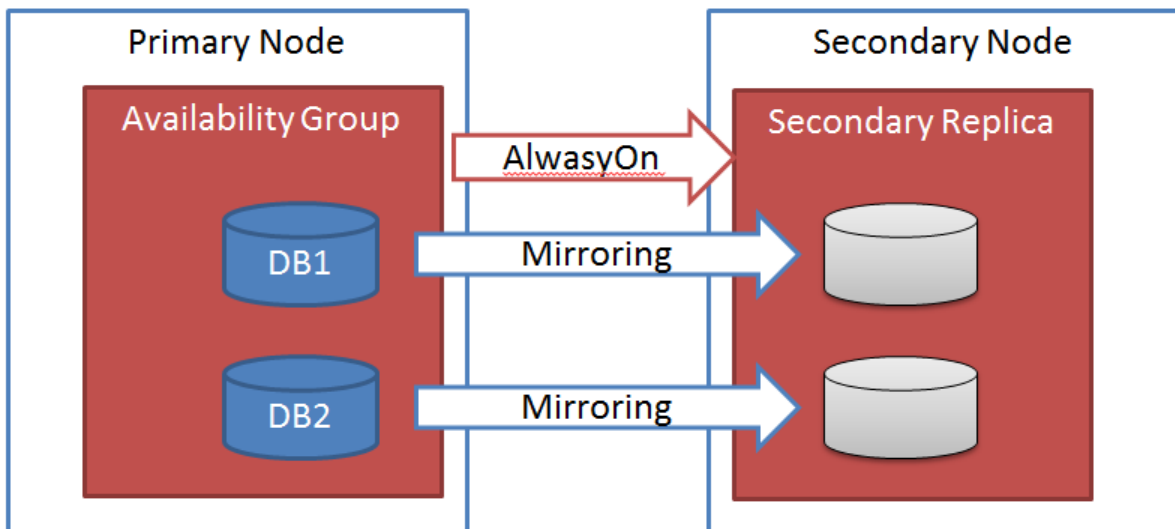
- Availability Groups : Databases with dependencies on one another fail over together, as a group.
- Multiple Secondaries : AlwaysOn will allow for multiple standby replicas for each availability group.
- Readable Secondaries : The standby replicas are accessible for read-only operations.

### Availability Groups

“데이터베이스 미러링(Database Mirroring)” 은 각 미러된 데이터베이스가 각각의 개체로 아래와 같이 운영됩니다. 그러나, 실제 어플리케이션들은 종종 하나의 SQL 인스턴스의 여러 개의 데이터베이스를 사용하도록 개발된 경우가 많습니다. 즉, 이러한 경우에 “데이터베이스 미러링(Database Mirroring)” 기술로 가용성을 높이기 위해서는, 실제 어플리케이션이 사용하는 데이터베이스 모두를 하나의 개체로 운영할 수 있는 방안이 제공되어야 합니다. 그러나, “데이터베이스 미러링(Database Mirroring)” 기술을 사용하면, 하나의 데이터베이스에 “장애 조치(failover)” 가 발생하면, 연관된 다른 데이터베이스들도 “장애 조치(failover)” 되도록 복잡한 로직을 별도로 작성해야 합니다.



그러나, "HADR"의 Availability Groups 기능을 사용하면, 이러한 고민을 해결할 수 있습니다. 즉, 아래와 같이 특정 어플리케이션에 연관된 모든 데이터베이스를 하나의 그룹으로 묶어서, 특정 하나의 데이터베이스에서 "장애 조치(failover)"가 발생하게 되면, 같은 그룹 내의 데이터베이스도 자동으로 "장애 조치(failover)" 되도록 할 수 있습니다.

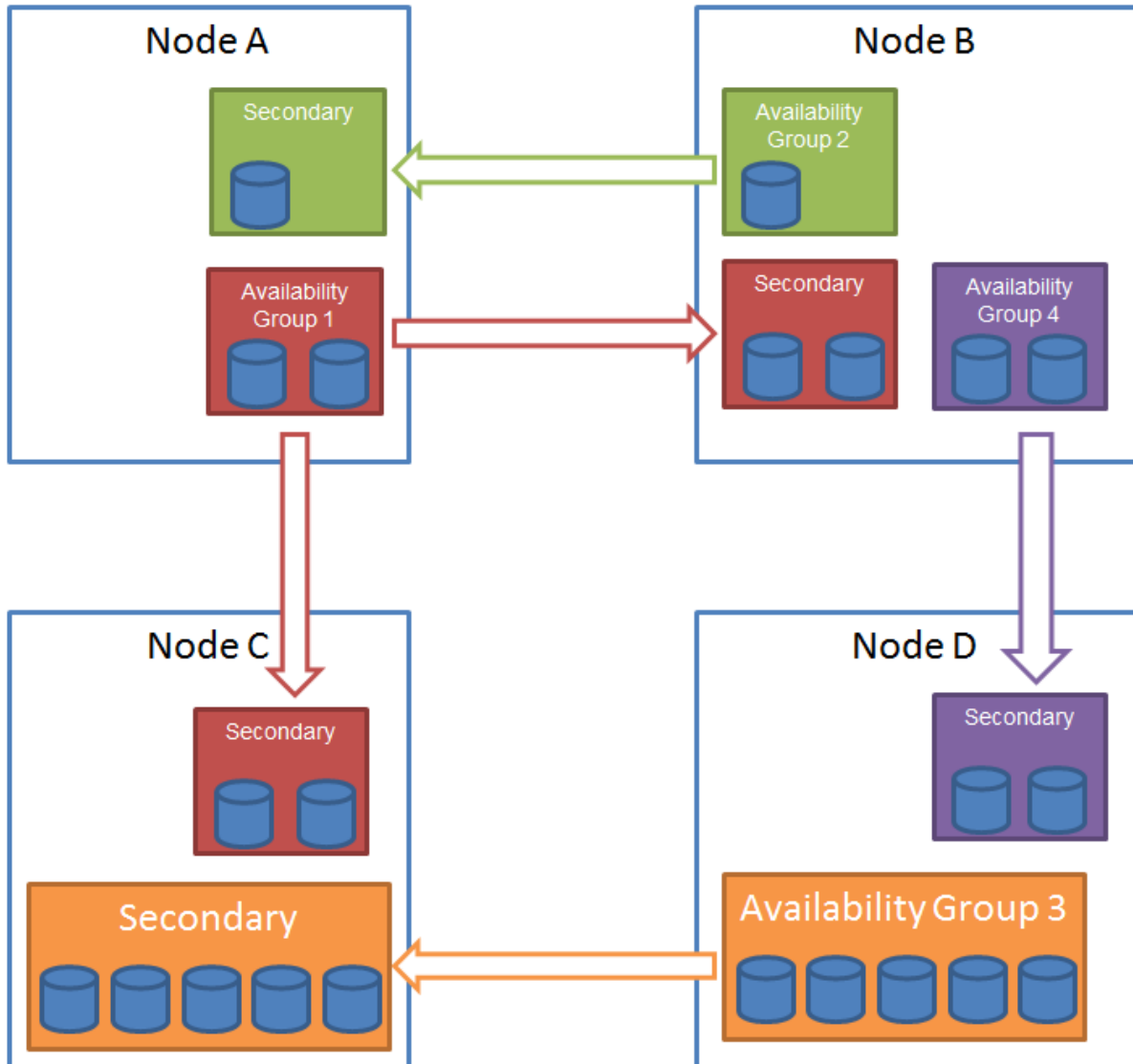


그러나, 여전히 Availability Groups 내의 모든 데이터베이스의 복제는 "데이터베이스 미러링(Database Mirroring)" 기술을 사용합니다.



## Multiple Secondaries

“데이터베이스 미러링(Database Mirroring)” 기술은 오로지 2 개의 SQL 호스트 사이에서만 구성할 수 있습니다. 그러나, 이러한 상황은 실제 운영 환경에서 2 개의 SQL 호스트가 동시에 문제가 발생할 경우에는 어플리케이션은 작동할 수 없습니다. 그러나, HADR 은 “데이터베이스 미러링(Database Mirroring)” 과는 달리 secondary replica 를 여러 개의 호스트에 구성할 수 있습니다. 아래는 multiple secondaries 를 구성한 예입니다. 총 4 대의 SQL 호스트를 사용하여 여러 개의 Availability Groups 에 대해서 다중 호스트에 secondary replica 를 구성한 예입니다.



- Availability Group 1 contains 2 databases and 3 nodes of the cluster have joined this availability group. The SQL Server instance on node A is the the primary replica and the ones nodes B and C each host an availability replica.

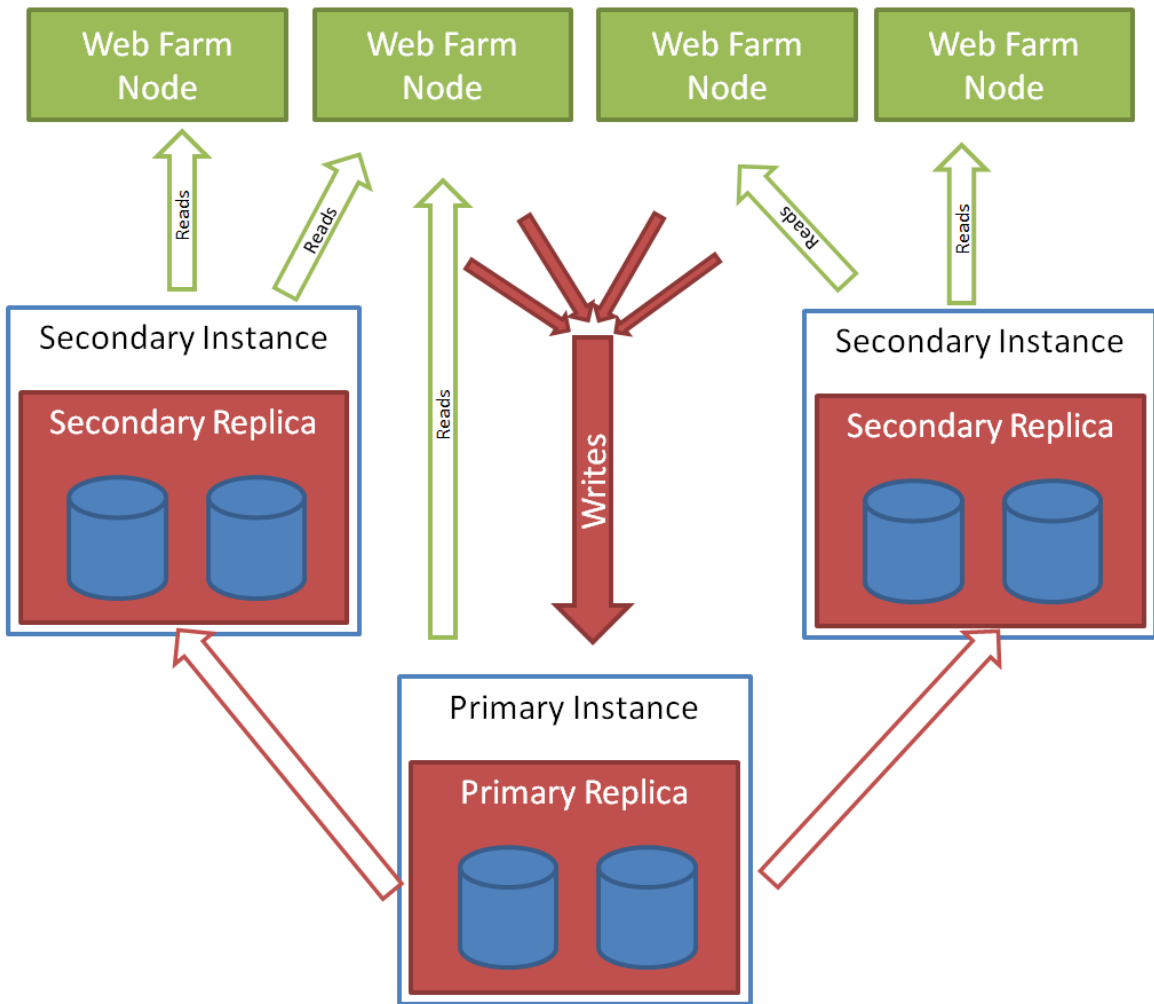
- Availability Group 2 contains one database, it has the primary availability group running on the SQL Server instance on node B of the cluster and an availability secondary replica on the instance on node A of the cluster.
- Availability Group 3 contains five databases, it has the primary availability group running on the SQL Server instance on node D of the cluster and an availability secondary replica on the instance on node C of the cluster.
- Availability Group 4 contains two databases, it has the primary availability group running on the SQL Server instance on node B of the cluster and an availability secondary replica on the instance on node D of the cluster.

위 4 대의 SQL 호스트는 동일 Windows Failover Clustering 내에 구성된 것을 가정합니다.

### Readable Secondaries

HADR 는 secondary replica 에 대해서 실시간 읽기 전용으로 구성할 수 있습니다. 두 번째 Availability Groups 내의 모든 데이터베이스에 대해서 읽기 전용 동작이 가능합니다. 이러한 기능을 잘 활용하면, 읽기 전용 어플리케이션에 대해서 HADR 은 Scale-Out 이 가능하도록 할 수 있습니다. "데이터베이스 미러링(Database Mirroring)" 스냅-샷 솔루션과는 달리, HADR 은 데이터베이스의 point-in-time 스냅-샷 뷰를 제공함으로써, 읽기 전용 secondary replica 는 실시간 읽기 쿼리를 수행할 수 있습니다. HADR 의 데이터베이스 point-in-time 스냅-샷 뷰 기능은 데이터베이스의 데이터 변경 사항을 실시간으로 반영할 수 있습니다. Secondary replica 에 대한 접근은 오로지 읽기만 가능하고, 어떠한 변경도 허용되지 않고, 모든 쿼리는 자동적으로 "snapshot isolation model" 방식으로 수행됩니다 (**당연히, lock hint 및 explicitly set isolation level 은 무시됩니다**). 이러한 방식으로 쿼리가 secondary replica 에 대해서 수행되기 때문에, 데이터의 일관성이 유지될 수 있습니다.

다음은 HADR 의 Scale-Out 읽기 전용 기능을 이용하여 웹 팜의 웹 서버들에게 확장성을 제공해 주는 하나의 예입니다.

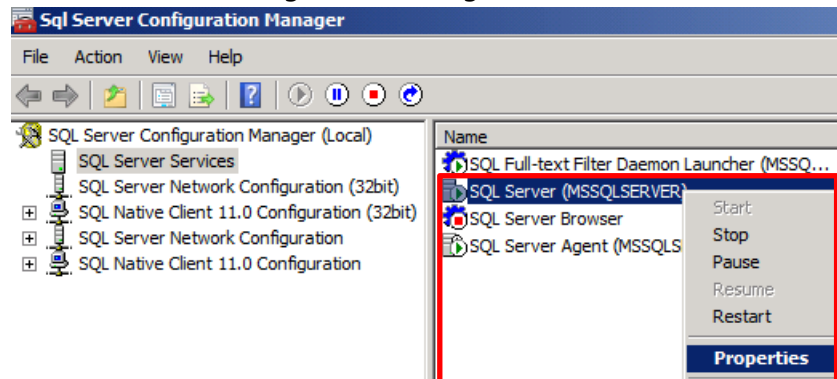


이상과 같이 HADR의 “데이터베이스 미러링(Database Mirroring)”과는 다른 기능에 대해서 살펴보았습니다. HADR은 “데이터베이스 미러링(Database Mirroring)”과는 완전히 다른 차원의 high availability를 제공합니다. 이제 “데이터베이스 미러링(Database Mirroring)”은 소규모 비즈니스에 적합한 것이고, HADR은 대규모 비즈니스의 high availability에 적합합니다. 물론, HADR은 구성하기 위해서는 Availability Groups에 참여하는 모든 노드가 WSFC(Windows Server Failover Clustering)에 참여해야 하는 제약이 있습니다. 이 점을 감안하더라도, HADR은 새로운 차원의 high availability를 제공합니다.

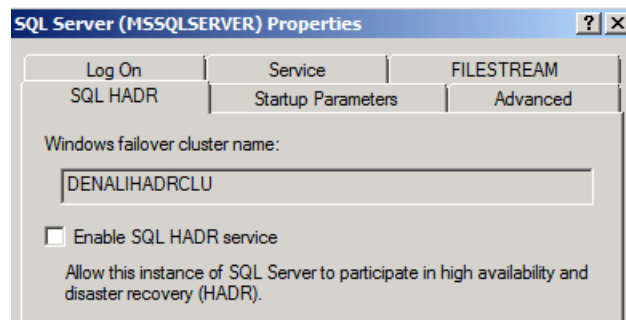
## HADR 구성에 참여할 노드의 SQL 인스턴스의 “HADR” 활성화

앞서 Failover Clustering 를 구축한 2 노드에 각기 Denali 의 로컬 단일 인스턴스로 SQL 설치를 완료했습니다. 이제 각 로컬 단일 인스턴스에 Denali 의 HADR 기능을 활성화 합니다. 이 과정은 HADR 기능을 이용하고자 하는 모든 SQL 인스턴스에서 수행해야 합니다. 다음 과정을 각각의 SQL 인스턴스에서 진행합니다.

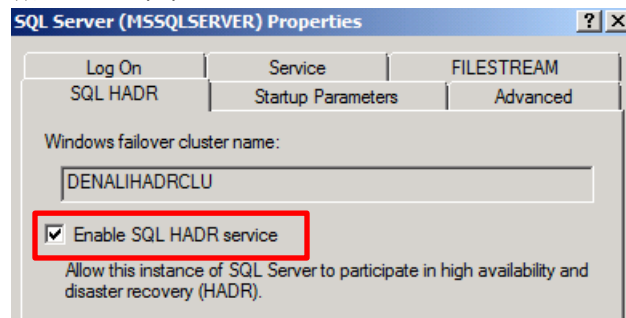
먼저, SQL Server Configuration Manager 도구를 사용하여, MSSQLSERVER 의 속성을 확인합니다.



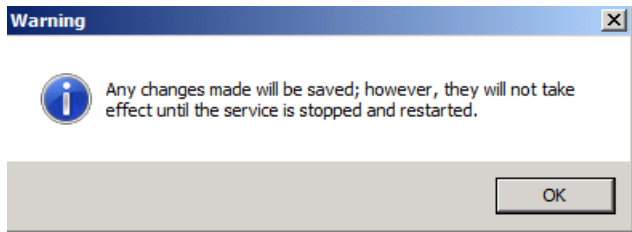
아래와 같이 “SQL HADR” 탭을 선택해 보면, 현재 SQL HADR 이 활성화되어 있지 않음을 알 수 있습니다.



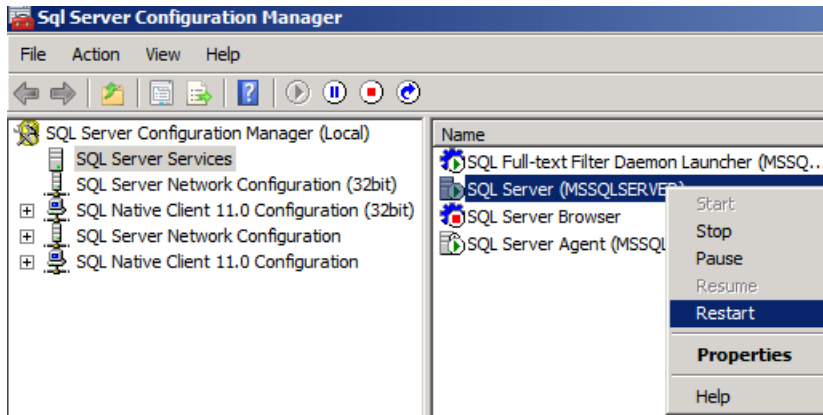
아래와 같이 “Enable SQL HADR service” 옵션을 선택합니다. 여기에서 주의 깊게 살펴볼 부분은 HADR 에서 사용할 Failover Clustering 에 대한 네트워크 이름(“DENALIHADRCLU”)이 정확하게 명시되어 있는 점입니다.



HADR 기능에 대한 활성화 속성을 저장에 되었고, 실제 SQL Service 를 재시작해야만, HADR 기능이 정상 작동할 수 있습니다.



아래와 같이 "MSSQLSERVER" 서비스를 재시작합니다.

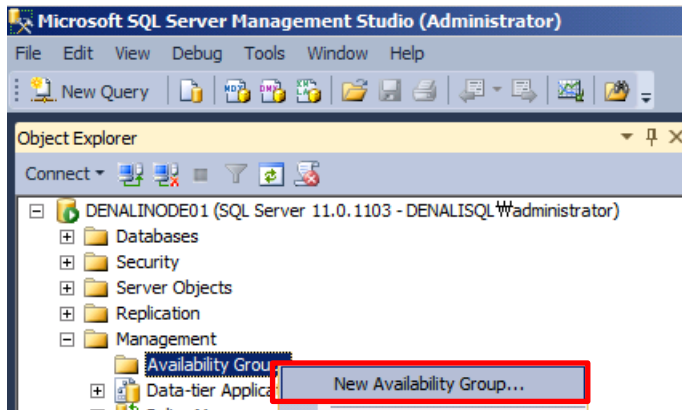


이와 같은 작업을 HADR 에 참여할 다른 노드에서도 동일하게 수행합니다.

## HADR 기능 테스트를 위한 Availability Group 생성

앞서 Failover Clustering 에 참여한 2 노드 (DENALINODE01 , DENALINODE03)에 설치된 각각의 SQL 단일 인스턴스에 HADR 기능을 활성화했습니다. 이제 HADR 기능을 사용할 사용자용 데이터베이스를 지정하여, Availability Group 을 생성하는 작업을 수행합니다.

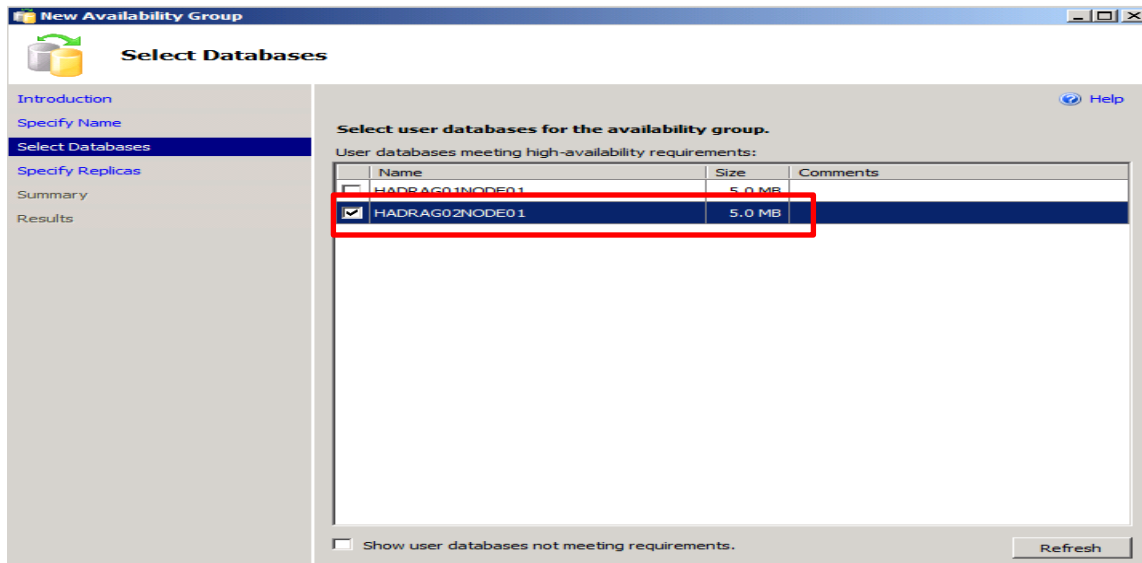
Availability Group 의 Primary Replica 역할을 수행할 "DENALINODE01" 서버의 SQL Management Studio 도구를 사용하여, Availability Group 을 생성합니다.



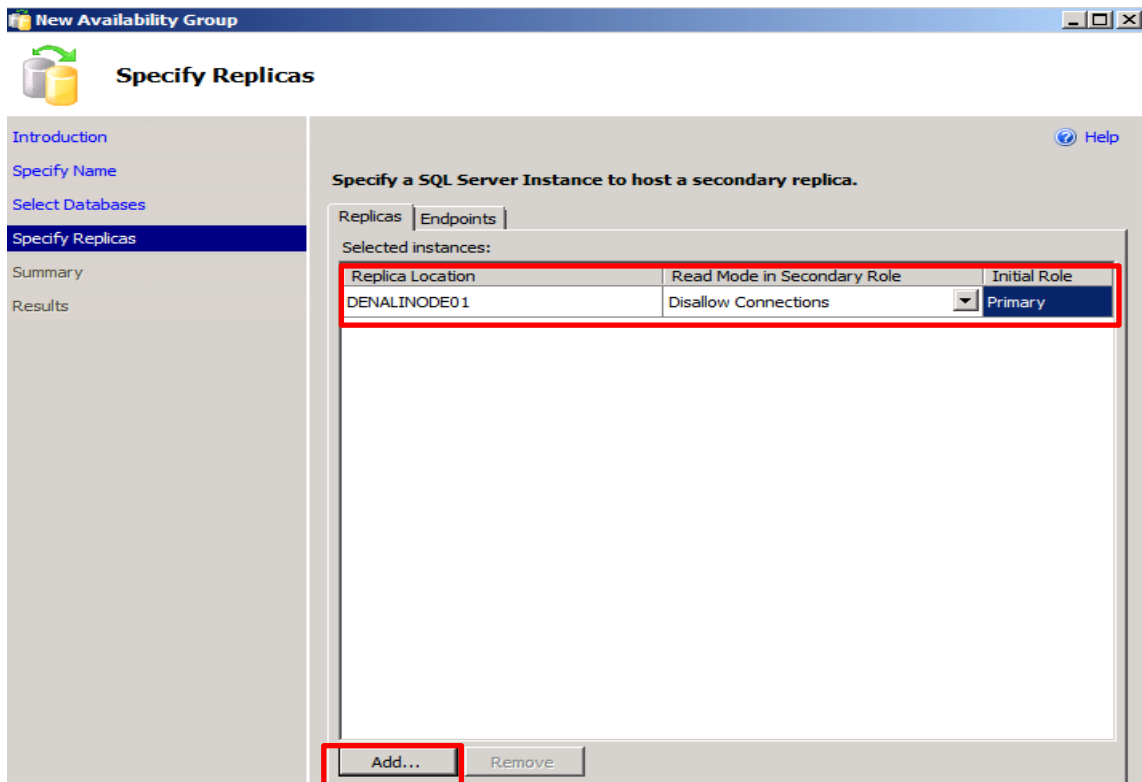
아래 단계에서 생성할 "Availability Group" 이름을 정합니다. 여기에서 지정한 이름이 추후 WSFC 에서 자원으로 등록됨을 유의합니다.



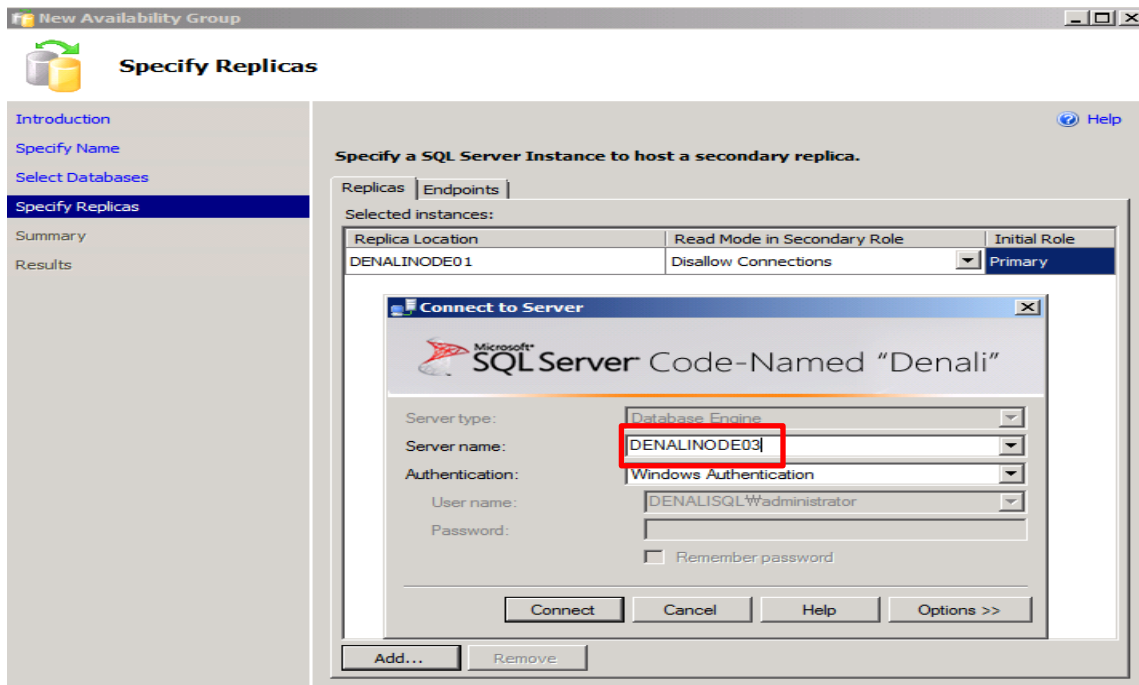
아래 단계에서 Availability Groups 에 포함될 사용자 데이터베이스를 지정합니다.



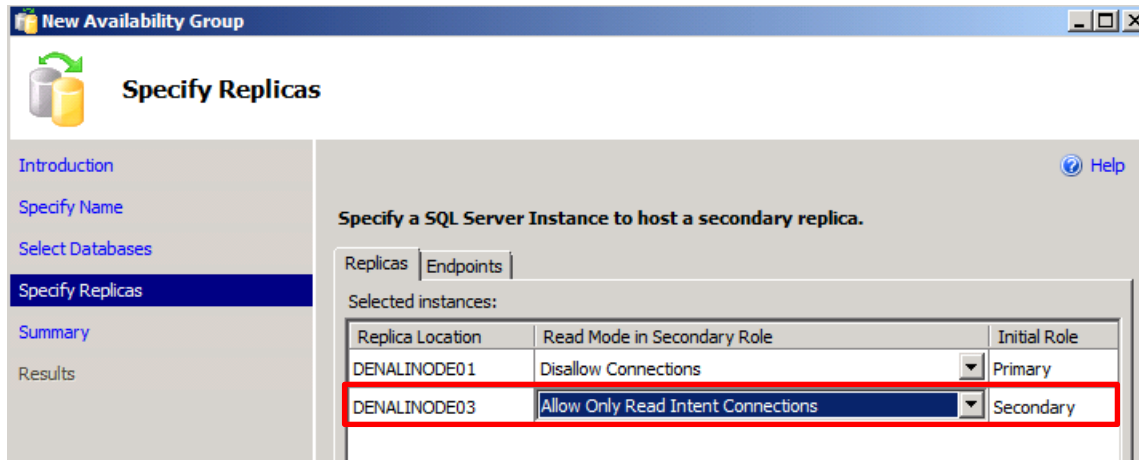
아래 단계에서 Secondary Replica 를 지정합니다. 기본적으로, 아래와 같이 "Primary Replica" 로써, 이 작업을 시작한 서버가 보여짐을 알 수 있습니다.



아래와 같이 Secondary Replica 를 지정합니다.

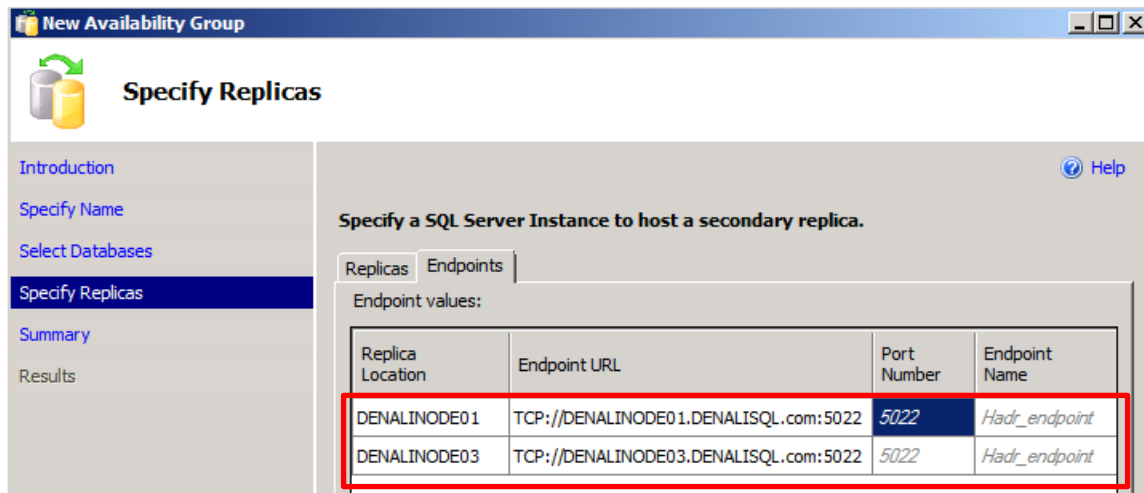


아래와 같이 Secondary Replica 에 대해서 "Allow Only Read Intent Connection" 모드로 설정합니다.

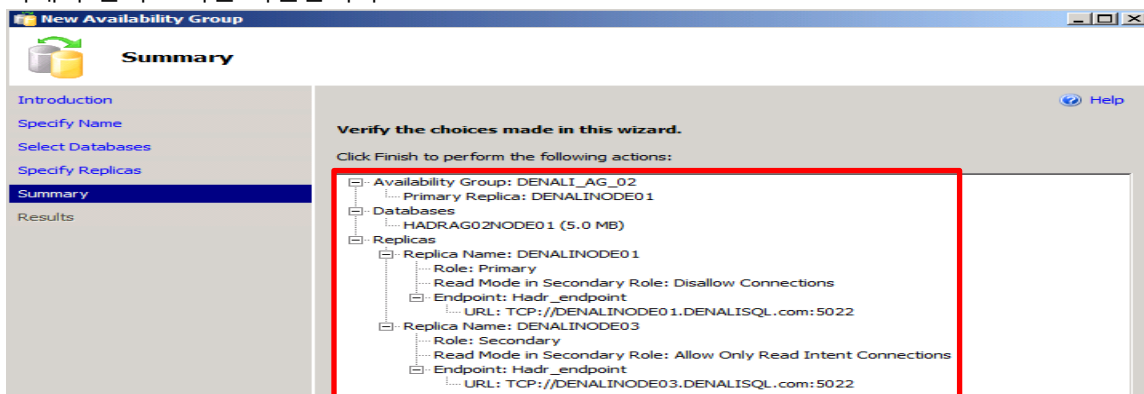


아래와 같이 자동적으로 생성되는 "Endpoint"를 확인합니다. 만약, 이 2 노드 사이에 방화벽이 존재한다면, 아래의 "5022" 포트에 대해서 방화벽 포트의 오픈이 필요합니다.

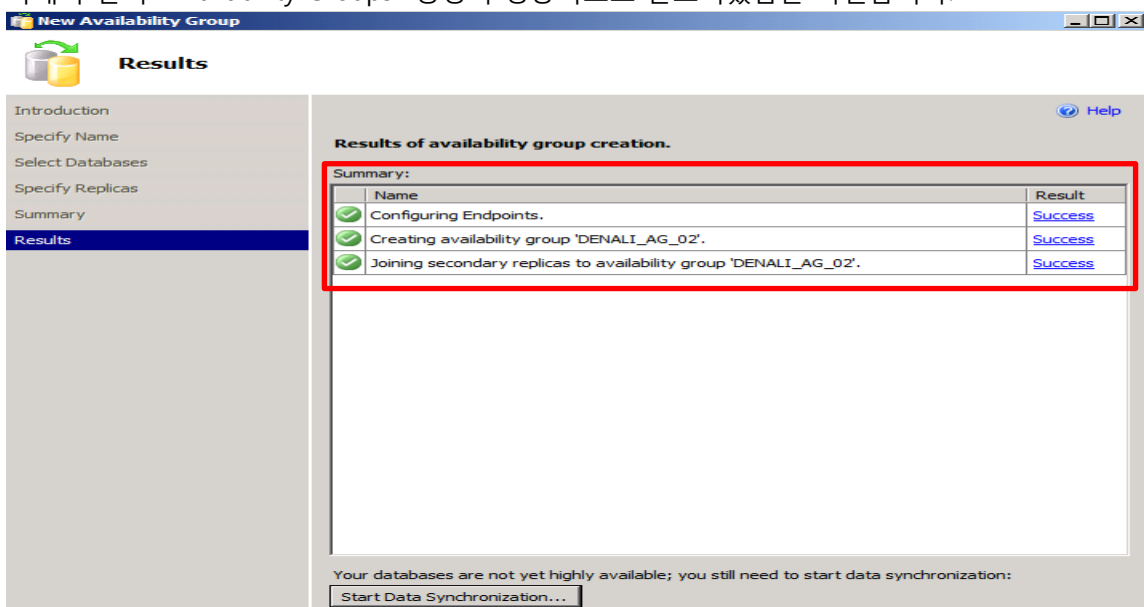




아래와 같이 요약을 확인합니다.



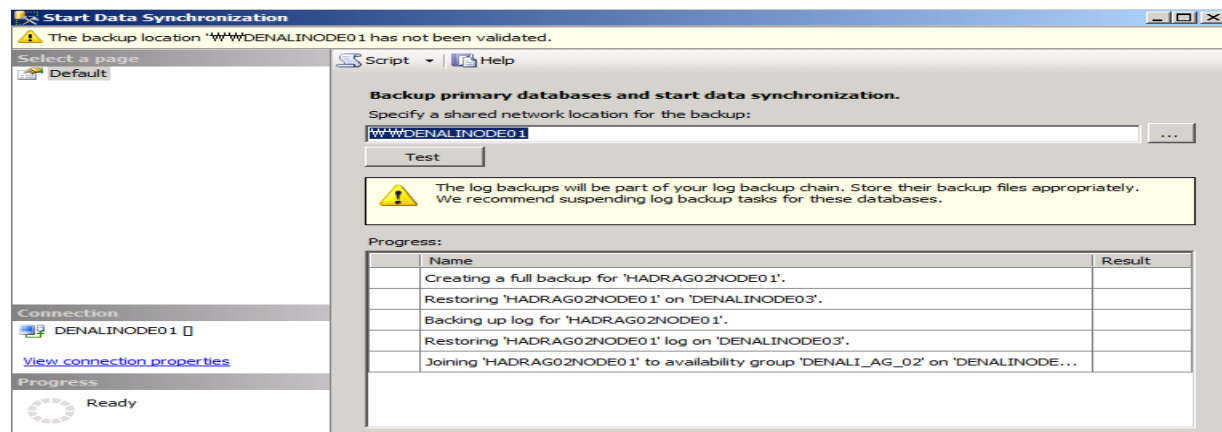
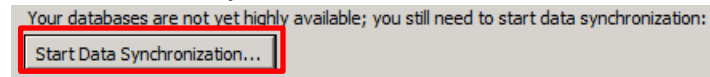
아래와 같이 "Availability Groups" 생성이 성공적으로 완료되었음을 확인합니다.



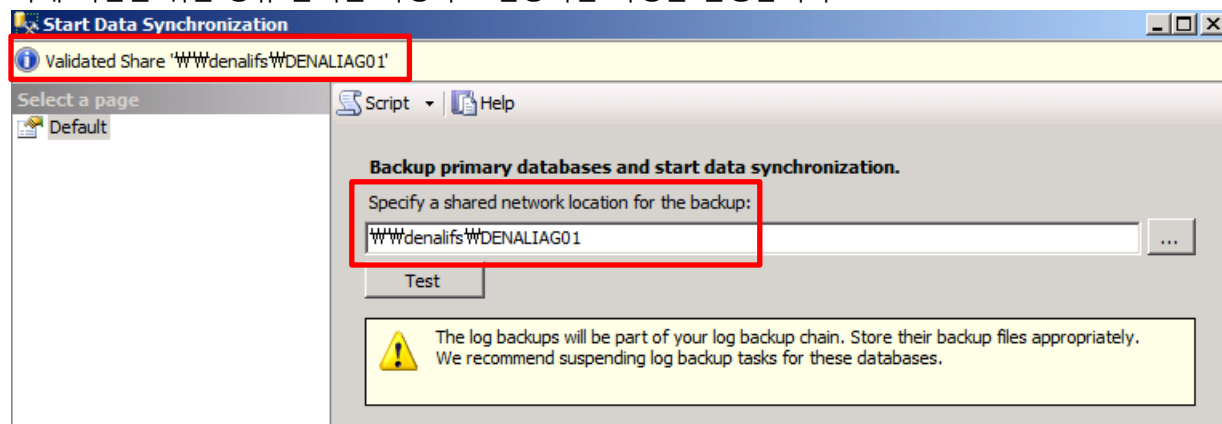
위와 같이 HADR 구현이 완료되었음을 확인합니다. 그러나, 아직 Primary Replica 와 Secondary Replica 사이의 초기 데이터 동기화 작업이 완료되지 않았습니다. SQL Code Name "Denali" CTP 버전에서는 별도의 데이터 동기화를 위한 GUI 메뉴를 제공하지 않으므로, 위 마법사를 닫지 말고, "Start Data Synchronization" 메뉴를 수행하여, 초기 데이터 동기화 작업을 진행합니다. 데이터 동기화를 위해서는 특정 서버의 공유 폴더가 사전에 구성되어 있어야 합니다. 이 공유 폴더에 적어도 로컬 administrators 그룹과 SQL 서비스 계정에 대해서는 "Full Control" 권한이 설정되어 있어야 합니다.

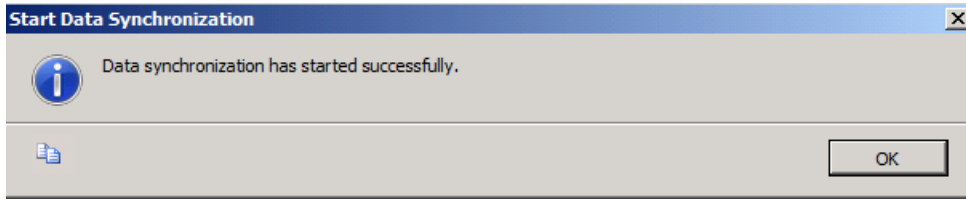
## HADR 의 Availability Group 의 데이터 동기화 작업

아래 "Start Data Synchronization" 메뉴를 수행하여 데이터 동기화 작업을 수행합니다.

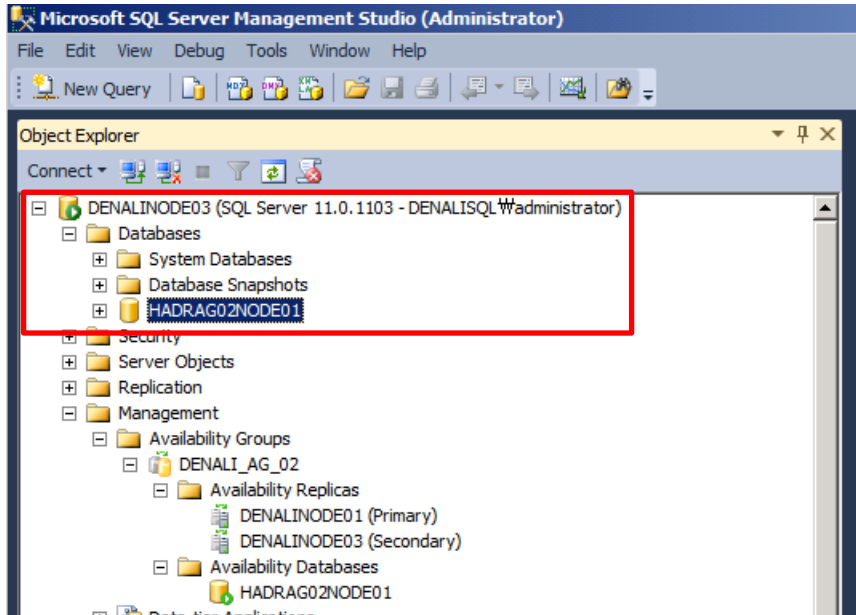


아래 백업을 위한 공유 폴더를 지정하고 검증하는 과정을 진행합니다.

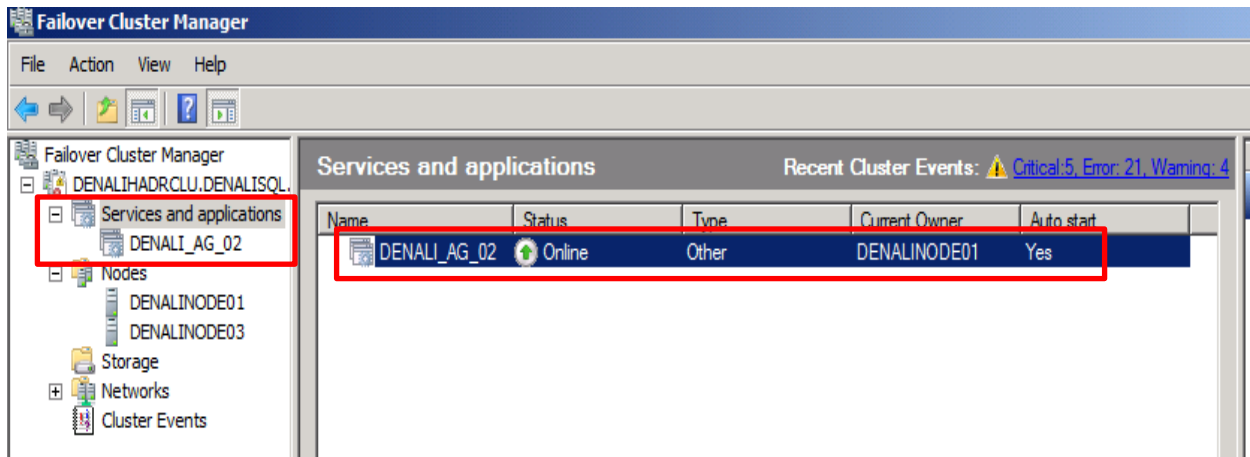




초기 데이터 동기화 작업이 완료된 후, 아래와 같이 SQL Management Studio 에서 확인해 보면, Primary Replica 및 Secondary Replica 를 확인 할 수 있고, 또한, "Availability Databases" 부분에서 Availability Group 에서 구성한 데이터베이스를 확인할 수 있습니다. Secondary Replica 역할을 수행하는 DENALINODE03 서버에는 아래와 같이 "HADRAG02NODE01" 이라는 사용자 데이터베이스가 생성되어 있음을 알 수 있습니다. 이 데이터베이스는 원래 DENALINODE01 서버에서 생성되었던 사용자 데이터베이스입니다.



위와 같이 HADR 구성 및 초기 데이터 동기화를 완료한 후, 2 노드의 Failover Clustering 관리자에서 확인해 보면, HADR 의 Failover Clustering 을 위한 "DENALI\_AG\_02" 라는 어플리케이션이 등록되어 있고, 그 어플리케이션 내부에는 "DENALI\_AG\_02" 라는 리소스가 "온라인" 되어 있음을 알 수 있다.



이상과 같이 첫 번째 "Availability Groups"의 생성 및 초기 데이터 동기화 작업까지 완료하였습니다.

## Primary Replica 에 연결하기 위한 Virtual Network Name 을 Failover

### Clustering 리소스 생성

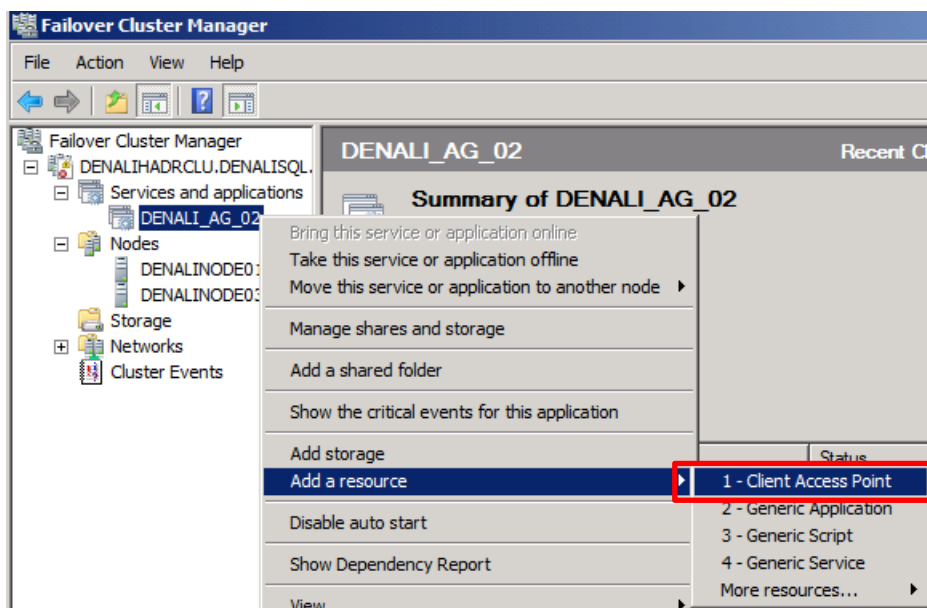
앞서와 같이 Availability Group 을 생성 및 구성 한 후에, 실제 어플리케이션은 현재 시점의 primary replica 에 항상 연결하기 위해서는 "Virtual Network Name(VNN)"을 생성함으로써 primary availability replica 에 연결할 수 있습니다. Availability Group 이 fail over 된 후에, VNN 은 새로운 현재 시점의 primary replica 를 연결할 수 있도록 합니다. 각 Availability Group 별로 도메인 내에서 unique 한 VNN 을 생성해야만 합니다.

실제 어플리케이션 내에서, TCP 프로토콜은 VNN 또는 VNN 에 해당되는 VIP 를 사용하여 Availability Group 에 연결할 수 있습니다. Availability Group 내에 있는 데이터베이스에 연결하는 데 사용하는 어플리케이션의 connection string 에서, 실제 서버 이름보다는 Availability Group VNN 을 지정합니다. 어플리케이션은 현재 시점의 Primary Replica 를 연결할 수 있습니다.

아래와 같이 connection string 을 사용할 수 있는 예제가 있습니다.

```
Server=tcp:MyVNN;Database=MyDB;IntegratedSecurity=SSPI  
Server=tcp:MyVNN,1433;Database=MyDB;IntegratedSecurity=SSPI
```

이 VNN 은 앞서 생성한 Availability Group 이 생성된 Failover Clustering 어플리케이션 내의 리소스로 등록합니다. 이 과정은 아래 단계를 수행하여 진행합니다.



**New Resource Wizard**

### Client Access Point

Enter Network Name and IP Address:

Name:

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
<input checked="" type="checkbox"/>	172.168.0.0/16	172.168.0.21
<input checked="" type="checkbox"/>	174.168.0.0/16	174.168.0.21

**New Resource Wizard**

### Confirmation

You are ready to make a Client Access Point.

**Network Name:** DENALIAG02VNN  
**IP Address:** 172.168.0.21  
**IP Address:** 174.168.0.21

To continue, click Next.

**New Resource Wizard**

### Summary

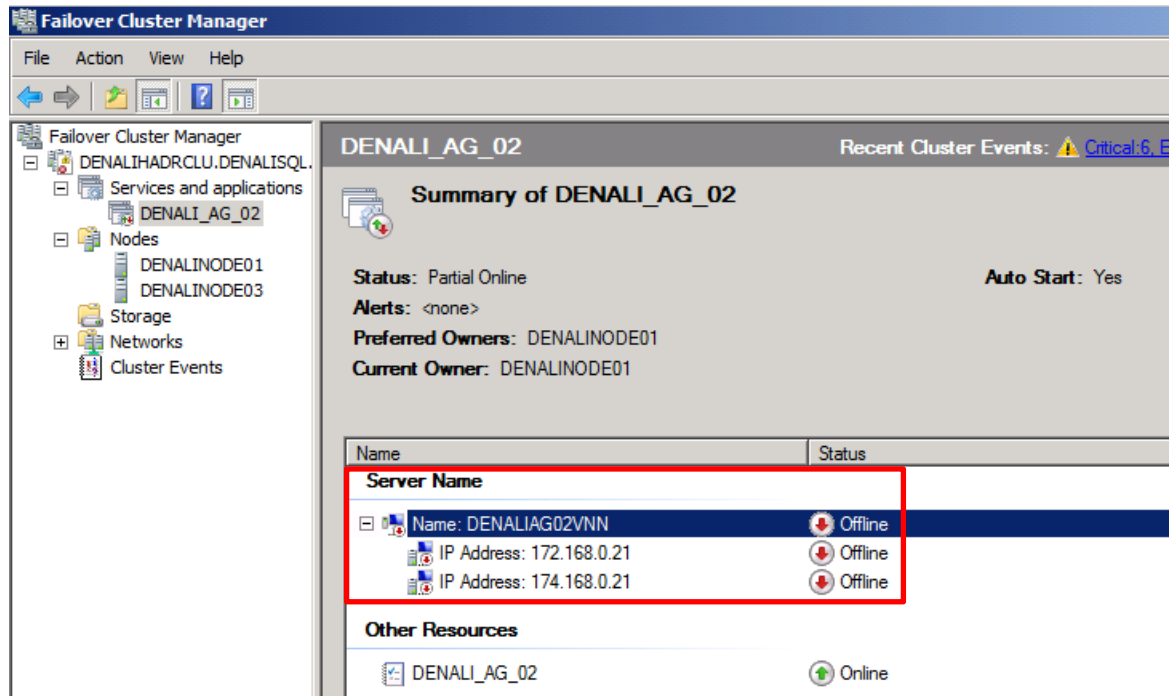
The new resource as was successfully created and configured.

**Resource:** Client Access Point  
**Network Name:** DENALIAG02VNN  
**IP Address:** 172.168.0.21  
**IP Address:** 174.168.0.21

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

[View Report...](#)

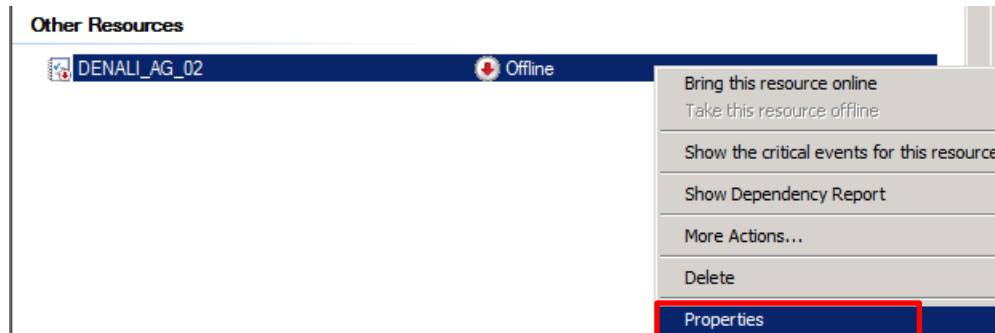
아래와 같이 Client Access Point 자원이 정상적으로 생성되었음을 확인합니다.



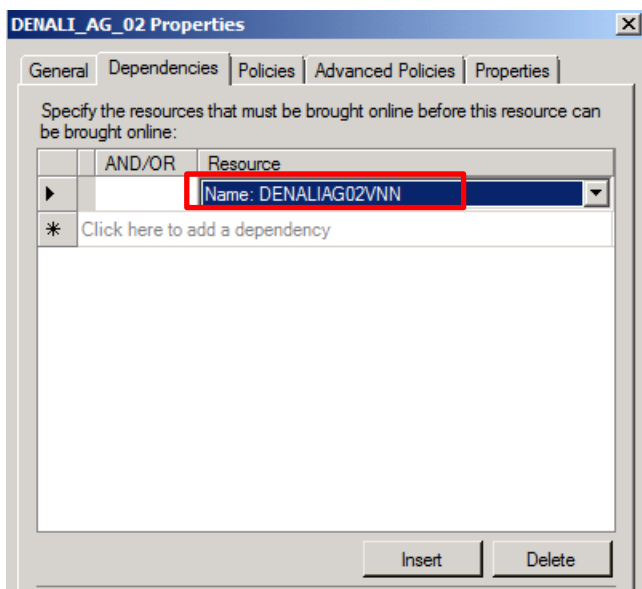
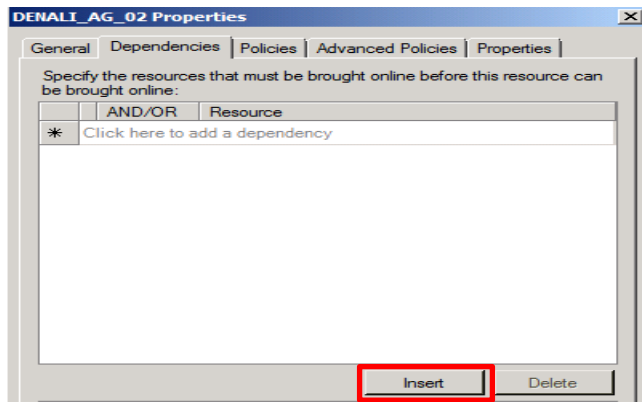
이제 종속성 설정을 위해 Availability Groups 자원인 “DENALI\_AG\_02” 을 오프라인합니다.



“DENALI\_AG\_02” 속성을 확인합니다.



아래 “Dependencies” 탭에서, 앞서 생성했던 “Client Access Point” 자원을 선택하여 추가하는 과정을 진행합니다.



이제 모든 자원을 온라인 시킵니다.

**Other Resources**



위와 같이 수행하면, 아래와 같이 생성한 VNN 을 확인할 수 있습니다.



**Failover Cluster Manager**

File Action View Help

Failover Cluster Manager

- DENALIHADRCLU.DENALISQL
  - Services and applications
    - DENALI\_AG\_02**
  - Nodes
    - DENALINODE01
    - DENALINODE03
  - Storage
  - Networks
  - Cluster Events

**DENALI\_AG\_02**      Recent Cluster Events: Critical 6, E

**Summary of DENALI\_AG\_02**

**Status:** Online      **Auto Start:** Yes

**Alerts:** <none>

**Preferred Owners:** DENALINODE01

**Current Owner:** DENALINODE01

Name	Status
<b>Server Name</b>	
Name: DENALIAG02VNN	Online
IP Address: 172.168.0.21	Online
IP Address: 174.168.0.21	Offline
<b>Other Resources</b>	
DENALI_AG_02	Online

## 테스트

AlwaysON 기능을 테스트하기 위해 아래와 같이 진행합니다. 사전에 "AdventureWorks" 데이터베이스를 생성하고 Availability Groups 으로 생성되었다고 가정합니다.

1. Create a simple table (can use below script) on "**PRIMARY Replica (DENALINODE01)**" and insert some rows (can use below script)

**Use AdventureWorks**

**GO**

```
CREATE TABLE [dbo].[New_Table](
```

```
  [ID][int] NULL,
```

```
  [NAME] [varchar](50) NULL
```

```
) ON [PRIMARY]
```

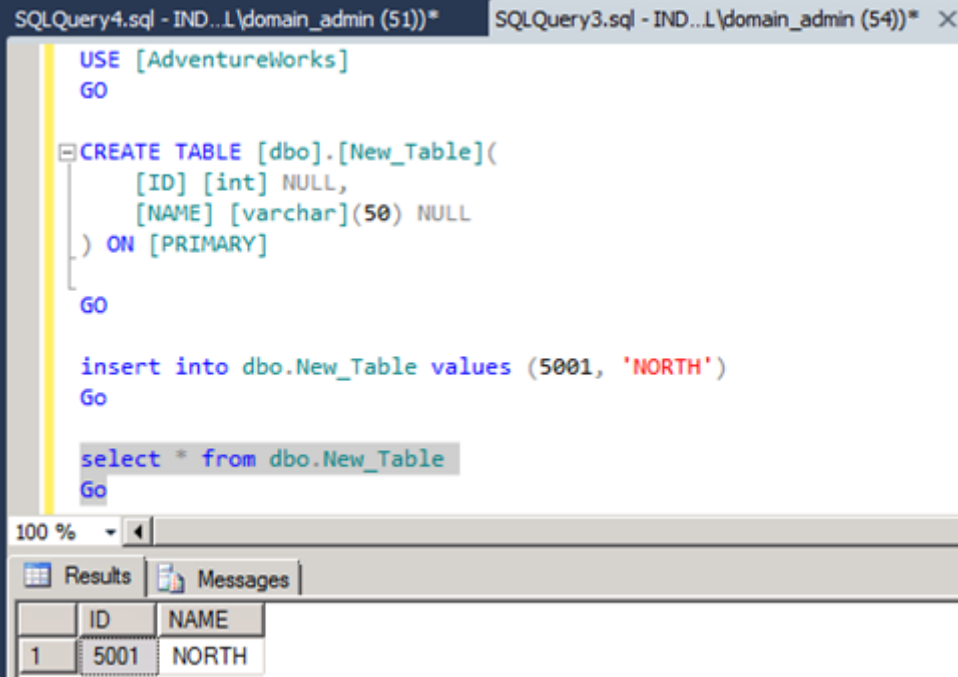
**GO**

```
INSERT INTO dbo.New_Table values (5001, 'NORTH')
```

**GO**

```
SELECT * from dbo.New_Table
```

**GO**



The screenshot shows a SQL Server query window with the following SQL code:

```
USE [AdventureWorks]
GO

CREATE TABLE [dbo].[New_Table](
  [ID] [int] NULL,
  [NAME] [varchar](50) NULL
) ON [PRIMARY]
GO

insert into dbo.New_Table values (5001, 'NORTH')
Go

select * from dbo.New_Table
Go
```

The results pane at the bottom shows a table with the following data:

	ID	NAME
1	5001	NORTH

2. Then, connect to "**SECONDRY Replica (DENALINODE03)**" and try selecting the rows. This will work!!



“장애 조치(failover)” 테스트는 이번 문서에서는 생략합니다.

## HADR 에 대한 DENALI CTP 1 의 Known Issue

아래와 같은 issue 들이 CTP 1 에서 알려진 문제점들입니다. 추후 개선될 것으로 생각됩니다.

- Can't Failover (no matter how hard I try)
- Only 2 Replica Servers are supported in CTP1
- There is no way to failover within SSMS. Powershell or Failover Cluster Manager is required.
- Testing the network share fails for no reason
- There is no Get-SqlAvailabilityGroup
- Needed for several of the PowerShell CmdLets
- No way to add databases or replicas via SSMS
- No way to add databases or replicas via SSMS
- No way to tell in Object Explorer if a database is an HADR database

## 참조 자료

Step-by-Step: Configuring a 2-node multi-site cluster on Windows Server 2008 R2 – Part 1  
(<http://clusteringformeremortals.com/2009/09/15/step-by-step-configuring-a-2-node-multi-site-cluster-on-windows-server-2008-r2-%E2%80%93-part-1/>)

DNS Registration with the Network Name Resource  
(<http://blogs.msdn.com/b/clustering/archive/2009/07/17/9836756.aspx>)

Configure Heartbeat and DNS Settings in a Multi-Site Failover Cluster  
([http://technet.microsoft.com/en-us/library/dd197562\(W.S.10\).aspx](http://technet.microsoft.com/en-us/library/dd197562(W.S.10).aspx) )

SQL Server Multi-Subnet Failover Clustering ([http://msdn.microsoft.com/en-us/library/ff878716\(v=SQL.110\).aspx](http://msdn.microsoft.com/en-us/library/ff878716(v=SQL.110).aspx))

Installing a SQL Server "Denali" Failover Cluster ([http://msdn.microsoft.com/en-us/library/ms179410\(v=SQL.110\).aspx](http://msdn.microsoft.com/en-us/library/ms179410(v=SQL.110).aspx))

How to: Create a New SQL Server Failover Cluster (Setup) ([http://msdn.microsoft.com/en-us/library/ms179530\(v=SQL.110\).aspx](http://msdn.microsoft.com/en-us/library/ms179530(v=SQL.110).aspx))

How to: Add or Remove Nodes in a SQL Server Failover Cluster (Setup)  
([http://msdn.microsoft.com/en-us/library/ms191545\(v=SQL.110\).aspx](http://msdn.microsoft.com/en-us/library/ms191545(v=SQL.110).aspx))

High Availability and Disaster Recovery ("HADR") (SQL Server) ([http://msdn.microsoft.com/en-us/library/ff878484\(v=SQL.110\).aspx](http://msdn.microsoft.com/en-us/library/ff878484(v=SQL.110).aspx))

"HADR" Deployment (SQL Server) ([http://msdn.microsoft.com/en-us/library/ff878265\(v=SQL.110\).aspx](http://msdn.microsoft.com/en-us/library/ff878265(v=SQL.110).aspx))

AlwaysOn: High-Availability and reads Scale-Out (<http://rusanu.com/2010/11/11/alwayson-high-availability-and-reads-scale-out/>)

SQL Server Denali - AlwaysON (HADR): Step-by-Setup setup guide  
(<http://blogs.msdn.com/b/sqlserverfaq/archive/2010/12/17/sql-server-denali-alwayson-hadr-step-by-setup-setup-guide.aspx>)