



# Windows Server® 2008

## Hyper-V™

### Hyper-V: Host to Host Failover Cluster Kurulum Dokümanı

Yayın Tarihi : Ağustos 2009  
Doküman Sürümü : 1.0  
Yazar : Serhat AKINCI



Serhat AKINCI – IT Pro  
<http://www.serhatakinci.com>  
[serhatakinci@gmail.com](mailto:serhatakinci@gmail.com)

## Doküman Hakkında

Bu doküman Windows Server 2008 Hyper-V platformunun failover cluster yeteneğini anlatmak için hazırlanmıştır.

Doküman içerisinde Hyper-V Failover Cluster yapılarındaki ihtiyaçların neler olduğu, bu ihtiyaçlar doğrultusunda yapılan seçimlerin yapı üzerindeki etkileri, temel storage bilgileri, cluster nodelarının ve diğer serverların topoloji içerisindeki konumları gibi teorik bilgilerin yanı sıra, storage ve nodeların yapılandırılması, failover cluster kurulumu, sanal makinelerin failover cluster yapısına dahil edilmesi gibi teknik konular da uygulamalı olarak ele alınmıştır.

Bu doküman ile organizasyonunuz içerisinde yer alan iki Hyper-V Host'u yazılım tabanlı bir iSCSI storage ile birlikte failover cluster yapısına dahil edip, sistemi fiziksel başarısızlıklara karşı koruma altına alabilecek, high available sanal makineler oluşturabileceksiniz. Bununla birlikte Quick Migration ve VM Failover gibi konuların nasıl gerçekleştiğini de öğrenmiş olacaksınız.

Umarım faydalı olur.

# İÇERİK

<b>1. Failover Cluster Nedir? .....</b>	<b>5</b>
<b>2. Fiber Channel ve iSCSI Storage'ler .....</b>	<b>7</b>
<b>3. Hyper-V Failover Cluster Yapılarında Storage'ın Önemi .....</b>	<b>9</b>
<b>4. Örnek Topoloji ve Temel Gereksinimler .....</b>	<b>11</b>
<b>5. iSCSI Storage Konfigürasyonu .....</b>	<b>14</b>
5.1. IP ve Domain Ayarları	14
5.2. Microsoft iSCSI Software Target Kurulumu	16
5.3. Microsoft iSCSI Software Target Üzerinde Virtual Disk Yaratmak	20
<b>6. Cluster Nodeların Konfigürasyonu .....</b>	<b>29</b>
6.1. IP ve Domain Ayarları	29
6.2. HV-Node1 Üzerine Gerekli Feature'ların Yüklenmesi	31
6.3. HV-Node2 Üzerine Gerekli Feature'ların Yüklenmesi	34
6.4. iSCSI Initiator ile HV-Node1 için iSCSI Storage Discovery	37
6.5. iSCSI Initiator ile HV-Node2 için iSCSI Storage Discovery	41
<b>7. Cluster için Disk Atamak .....</b>	<b>45</b>
7.1. iSCSI Storage üzerinde HV-Node1 için Target Tanımı	45
7.2. iSCSI Storage üzerinde HV-Node2 için Target Tanımı	49
7.3. iSCSI Storage üzerinde HV-Node1 için Disk (virtual disk) Ekleme	54
7.4. iSCSI Storage üzerinde HV-Node2 için Disk (virtual disk) Ekleme	57
7.5. HV-Node1 üzerinde Disk (virtual disk)'i Görmek	60
7.6. HV-Node2 üzerinde Disk (virtual disk)'i Görmek	66

<b>8. Failover Cluster Kurulumu .....</b>	<b>69</b>
8.1. Cluster Kurulumu Öncesi Kontrol	69
8.2. Cluster Kurulumu	77
8.3. Quorum Disk Konfigürasyonu	85
<b>9. Hyper-V Sanal Makinelerini Cluster Yapısına Dahil Etmek .....</b>	<b>95</b>
9.1. Nodelar Üzerine Hyper-V Kurulumu	96
9.2. Test için Sanal Makine (VM) Yaratılması	98
9.3. Failover Cluster Yapısına Virtual Machine Servisini Ekleme	106
<b>10. Hyper-V Failover Cluster Sonrası Gelen Yetenekler .....</b>	<b>118</b>
10.1. Quick Migration - QM (Hızlı Taşıma)	118
10.1.1. Quick Migration Nedir?	118
10.1.2. Quick Migration Nasıl Yapılır?	120
10.2. VM Failover (Sanal Makine Hata Telafisi)	129
10.2.1. VM Failover Nedir?	129
10.2.2. VM Failover Nasıl Yapılır?	130
<b>11. Diğer Kaynaklar .....</b>	<b>140</b>

## 1. Failover Cluster Nedir?

Failover Cluster (Küme) teknolojisi (servisi), organizasyonumuz içerisindeki sunucuların ve bu sunucular üzerinde çalışan çeşitli uygulamaların/servislerin fail olması durumunda (genelde donanımsal başarısızlıklar yüzünden), hizmet veremez duruma geçen ilgili servis ve uygulamaların cluster üyesi diğer sunucu/sunucular üzerinde hizmet vermeye devam etmesini sağlayan bir teknolojidir.

Anlamalı bir Failover Cluster yapısı için genelde fiziksel sunucular ile temsil edilen (bazı senaryolarda sanal da olabilir) ve node olarak adlandırılan en az iki sunucu olmak zorundadır. Bununla birlikte çoğu zaman node'ların erişebildiği ortak bir storage (hardware/software) şartı da vardır.

Cluster yapıları **mission-critical** olarak tanımlanan birçok uygulama ve servis için önemlidir ancak sanallaştırma tarafındaki önemi çok daha büyüktür.

Sanallaştırmanın temel amaçlarından olan konsolidasyon yani çok sayıdaki fiziksel sunucuyu daha az sayıdaki fiziksel sunucu üzerinde sanal olarak konumlandırma işlemi sonucunda, aynı fiziksel host üzerinde birden fazla sanal iş yükü çalışmaya başlar ve host'un down olması durumunda üzerindeki tüm iş yükleri bu durumdan olumsuz etkilenir. Yani cluster ile güvenceye alınmamış ortamlarda çalışan sanal sistemlerde risk çok daha büyüktür.

Cluster yapılarıyla amaçlanan ise, beklenmedik bir anda meydana gelen arızalara ve genelde donanımsal sorunlar yüzünden yaşanan servis kesintilerine karşı önlem almak, bu doğrultuda ilgili uygulamaları/servisleri otomatik olarak diğer cluster üyesi sunucular üzerinden hizmete sunabilmektir.

Konuyu basit bir örnekle biraz daha açmak istiyorum.

İlk senaryomuzda sanallaştırma olmayan bir ortamda çalıştığımızı düşüyoruz ve organizasyonumuz içerisinde 5 fiziksel sunucu bulunuyor.

Bu sunucuların rolleri ise şu şekilde:

1. DC + DNS + DHCP (Domain Controller)
2. SQL Server
3. Exchange Server
4. ISA Server
5. Application Server (üzerinde herhangi bir ticari uygulama olabilir)

Bu ortamdaki fiziksel sunuculardan birisi herhangi bir nedenden dolayı down olduğu zaman, üzerinde çalışan uygulamalar ve servisler de fail olur ve hizmet veremez.

Örneğin SQL Server makinesinin down olduğunu düşünelim. Eğer SQL Server için oluşturulmuş cluster senaryonuz yoksa (ki bizim örnek yapımızda yok), bu durumda ilgili DB'lere erişi durur ve SQL Serverı ayağa kaldırmak ya da DB'leri başka bir sunucu üzerinde online hale getirmek için stresli bir süreç başlar. Organizasyon içerisindeki diğer sunucular ise kendi işlerini yapmaya devam edebilirler. Örneğin DC

hala ayakta'dır, mail trafiđi devam ediyordur, internet eriřimi vardır vs.. Sorun yapının sadece bir b3l3m3ne etki etmiřtir.

İkinci senaryomuzda ise ilk senaryodaki fiziksel ortamı bařarılı bir konsolidasyon s3reci sonunda tek bir Host 3zerinde sanallařtırdıđımızı d3ř3nelim. Yani artık elimizde 5 fiziksel sunucu yerine tek ve daha g3çl3 bir fiziksel sunucu var. 3zerinde ise 5 ayrı sanal makine (VM yada sanal iř y3k3) alıřıyor.

Fiziksel - Windows Server 2008 Hyper-V Host  
Sanal - DC + DNS + DHCP (Domain Controller)  
Sanal - SQL Server  
Sanal - Exchange Server  
Sanal - ISA Server  
Sanal - Application Server (ticari bir uygulama olabilir)

G3rd3đ3n3z gibi aynı hizmetler sanal olarak alıřmaya devam ediyor. Derken host (fiziksel) sistem donanımsal bir sorunla karřılařıyor ve down oluyor. İřte bu an felaketin bařladıđı andır 3nk3 host sistem ile birlikte 3zerindeki t3m sanal makineler (sanal iř y3kleri) de fail olmuřtur. Yani 5 sanal sunucu ve 3zerindeki uygulamalar/servisler hizmet veremez duruma gemiřtir. Bir anda firmanın t3m iřleri durur ve o an iin hızlı bir geri d3n3ř senaryonuz yoksa sanal makineleri ayađa kaldırmak olduka zaman alabilir. Bu durumda yařanan stres ve durumun iř s3relerine etkisi tamamı fiziksel olarak alıřan ortamdan ok ok daha fazladır.

Toparlarsak; tamamı fiziksel olarak alıřan yapılardaki donanımsal bařarısızlık genelde down olan tek bir sunucuyu ve 3zerindeki uygulamaları/servisleri etkiler (dolaylı yoldan farklı etkileri olabilir). Ama sanallařtırılmıř ve 3nlem alınmamıř ortamlardaki donanımsal bařarısızlıklar birden fazla sanal iř y3k3n3 etkiler ve sonuları ok daha fazla hissedilir.

Sanallařtırılmıř ortamlarda bu gibi felaketlerin 3n3ne geebilmek adına Hyper-V teknolojisi Windows Server Cluster servisini kullanabilme yeteneđine sahiptir ve en az iki fiziksel host ile k3melenerek bařarısızlık durumunda failover (hata telafisi) sađlayabilmektedir.

Failover Cluster yapıları temelde mission-critical uygulama ve servisler iin downtime'ı en aza indirmeyi amalar ve cluster 3yesi node'lar 3zerindeki uygulama ve servisler iin High Availability – HA (Y3ksek Eriřilebilirlik) sađlar. Hyper-V 3zerinde alıřan sanal makineler ise Windows Failover Cluster servisi ile High Available řekilde alıřabilirler.

## 2. Fiber Channel ve iSCSI Storage'ler

Failover Cluster özel bir teknoloji olduğu için yazılımsal ve donanımsal anlamda bir takım ihtiyaçları vardır. Özellikle storage tarafında seçeceğimiz bağlantı tipi (iSCSI/FC), disk özellikleri (SCSI, SAS, Sata vs..) ve ürünün donanımsal yada yazılımsal olması yapının maliyetine ve performansına direkt olarak etki eden faktörlerdir.

Storage aslında depolama anlamına gelmektedir. Örneğin server üzerindeki local bir diski de storage olarak adlandırabiliriz. Bizim cluster yapılarında bahsettiğimiz storage ise serverların ortak olarak erişebildiği depolama alanları sağlayan cihazlar/yazılımlardır. Bu nedenle yazıda geçen storage kelimesini ortak depolama alanı olarak algılamalısınız (ör: SAN).

**FC** yani **Fiber Channel** bağlantılı storage'lerin sahip olma ve ekipman maliyetleri iSCSI storage'lere göre daha yüksektir. Bununla birlikte FC storage'ler iSCSI'e göre daha performanslıdır.

Bu gün 4 ya da 8GB'lık FC HBA'ler var ve bu HBA'ler dual olarak çalışabiliyor. Yani 8Gb x 2 şeklinde. Ayrıca bu yapılarda serverlar ile storage arasındaki bağlantı fiber optik kablolar ile gerçekleşiyor ve bu ürünler uygun disk'ler ile desteklendiği taktirde çok yüksek hızlarda okuma/yazma (disk i/o) değerleri sağlayabiliyor.

**iSCSI** ise network (ethernet) üzerinden çalıştığı için storage erişim hızı network bandwidth ile doğru orantılı durumdadır. Yaygın olarak kullanılan hız ise 1Gbps dir. 10Gbps NIC ve network donanımları da kullanılabilmektedir ancak bu donanımlar şimdilik 1Gbps ürünler kadar yaygın değiller. Ayrıca ethernet teknolojisinin gelecekteki hali olan 100Gbps üzerinde çalışmalar devam ediyor ve bu teknoloji şu an çalışır durumda. Yılların teknolojisi ethernet ilerleyen senelerde de hız kesmeden yoluna devam edecek ve bu gelişim iSCSI bağlantılı storage'ler açısından oldukça iyi olacak gibi görünüyor.

Bağlantı tipi ve hızı dışında okuma/yazma performansını etkileyen bir diğer nokta ise storage üzerindeki disklerdir. Genelde Sata, SAS, SCSI diskler kullanılır ve bu diskler her iki bağlantı teknolojisi için de kullanılabilmektedir. Bu fiziksel disklerin rpm değerleri ve bağlı bulundukları kontroller kartların donanımsal özellikleri de okuma/yazma performansına direkt etki etmektedir. Bu gibi çok sayıda fiziksel diski bulunan, bu disklerin çeşitli bus yapıları ile birbirine bağlı olduğu storage'ler genelde SAN (Storage Area Network) olarak adlandırılırlar.

İnsanların kafasında Fiber Channel = SAN şeklinde bir eşleşme yer etmiş durumda ama bu doğru değil çünkü SAN bir topolojidir. Protokolü fiber channel olabileceği gibi TCP/IP de olabilir.

Tüm bunların dışında storage üzerinde konumlanacak uygulamanın tipi ve disk kullanım karakteristiği de performans ve storage seçimi açısından önemli kriterlerdir. Sanallaştırma ortamlarınızda yoğun çalışan VM'leriniz varsa ve maliyeti karşılayabiliyorsanız, her zaman için Fiber Channel storage'ler kullanmanızı öneririm.

Eğer FC arabirimi olan bir storage kullanmayı planlıyorsanız, storage ve diskler dışında temin edilmesi gereken bazı ek donanımlar da vardır. Fiziksel sunucuların Örneğin Serverların FC storage'e erişebilmesi ve üzerindeki LUN'ları kullanabilmesi için gerekli HBA'ler (Server'lara Fiber Optik arabirim sağlar) ve ihtiyaç halinde SAN switch'ler ek donanımlar için verilebilecek en güzel örnekleridir. Bu donanımlar tabi ki maliyete etki eder.

iSCSI arabirimli storage'lara ise daha düşük maliyetler ile sahip olmak mümkün. Yine serverların iSCSI storage'lere erişmesi için kullanılabilecek özel HBA'ler var ancak bunların maliyeti FC HBA'ler kadar yüksek değil. Ayrıca sıradan network kartlar ile de iSCSI Storage'lere erişmek mümkün. Ama yukarıda da bahsettiğimiz gibi performans Fiber Channel kadar yüksek olmayacaktır.

**(FC) Fiber Channel:** Aslında bir protokoldür. Bizi ilgilendiren taraftan baktığımızda ise fiber optik bağlantı arabirimi olduğunu söyleyebiliriz. Yani ethernet gibi bir arabirim. Kabaca server ile storage arasında çekilmiş fiber optik bir kablo (gerekli durumlarda arada bir fiber switch olabilir) ve bu kablonun server ve storage üzerinde uygun FC HBA'ler ile sonlandırıldığı sistemlerdir. Server üzerinde tanımlanan storage birimlerine (LUN) iletilmesi gereken disk i/o requestlerinin, yine bu kanal yani fiber optik hat üzerinden geçerek uygun protokol ile iletildiği yapılar olarak düşünebilirsiniz.

**(iSCSI) Internet Small Computer System Interface:** iSCSI de bir protokoldür ve temel amacı SCSI paketlerini TCP/IP networkleri üzerinde taşımaktır. İsmi başındaki (i) de buradan gelmektedir. iSCSI sistemleri zaten var olan bir teknoloji ile başka bir teknolojinin birleşimi gibi düşünebilirsiniz. SCSI paketlerini alıyoruz, ip trafiği üzerine bindiriyoruz... Ama performans FC'e göre daha düşüktür çünkü TCP/IP yüksek blok i/o için tasarlanmış bir protokol değildir ve normal bir networkte (ethernet) max. 1512kb frame size sahip paketler taşıyabilir, yani yaklaşık 1,5kb blok i/o yapılabilir. TCP/IP mimarisi gereği ortaya çıkan bu durum iSCSI tarafındaki verimi düşüren en önemli etkidir. Günümüzde yaygınlaşan ve Windows Server 2008 R2'nin de desteklediği Jumbo Frame özelliği sayesinde TCP/IP üzerinde yaklaşık 8-9kb'lık blok i/o yapılabilir. Tabi bu özelliğinin işletim sistemi, ethernet kartları ve switch'ler tarafından destekleniyor olması gerekir.

Peki, storage'in ve bağlantı teknolojisinin Hyper-V Cluster senaryolarındaki önemi nedir?

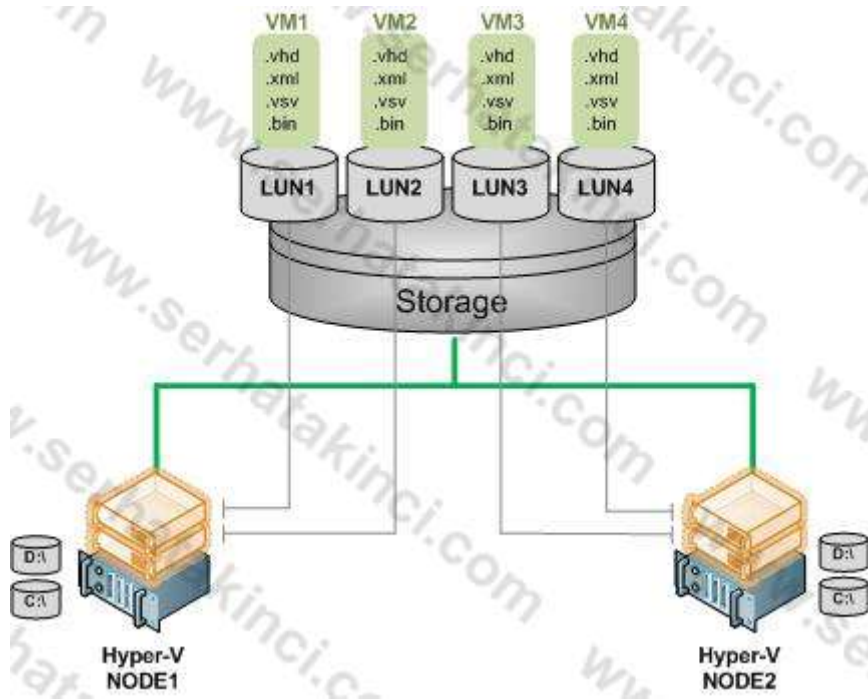


### 3. Hyper-V Failover Cluster Yapılarında Storage'ın Önemi

Host to Host Failover Cluster senaryosunda VM'ler (sanal makineler) host sistemler üzerinde değil, her iki node'un da erişebildiği ortak bir storage üzerinde durur. Burada storage üzerinde duran VM'den kastımız VM'in background dosyalarıdır. Yani VM'ler host'lar üzerinde çalışıyor ama XML, VHD, VSV, BIN gibi ilgili dosyaları storage üzerinde duruyor.

Aşağıda örnek bir topology var.

Buradaki storage FC yada iSCSI olabilir. Her iki fiziksel host -ki biz bunları node olarak adlandırıyoruz- storage'e üzerindeki ilgili birimlere (LUN) erişebiliyor. Storage üzerinde 4 ayrı LUN var ve her bir LUN içerisinde ayrı bir VM duruyor. Bu 4 VM, 2 Hyper-V Host'u üzerinde çalışıyor.



Bu yapıda storage'in önemi ise şu: Node'lar üzerinde çalışan ama arka planda storage üzerinde konumlanan VM'ler işletim sistemi(sanal) ve uygulama(sanal) process'lerini gerçekleştirdikçe, ilgili veriler storage üzerindeki ilgili dosyalara yazılır (VHD, BIN, VSV vs..). Bu durumda storage bağlantı teknolojisi ve cihaz üzerindeki diskler ne kadar hızlı ise, bu işlemler de o kadar rahat ve hızlı gerçekleşir ve bu doğrultuda VM'ler o derece performanslı çalışır.

Örneğin elimizdeki storage'in iSCSI arabirimi kullandığını düşünelim ve 1gbps hızında bir networkümüz olsun. Bu durumda storage erişim hızı max. 1gbps ile sınırlıdır. Yani elimizde sanal makine aksiyonlarına ait verilerin aktığı tek şeritli bir yol varmış gibi düşünebilirsiniz. Ama storage arabirimi olarak Fiber Channel kullanmış ve örneğin yapıyı 8gbps HBA'ler ile kurmuşsanız, elinizde 8 şeritli bir otoban olduğunu söyleyebiliriz. Ayrıca desteklendiği durumlarda dual HBA kullanmanız demek, 8gbps x 2 şeklinde 16 şeritli bir otoban demek ☺

Sonuç olarak host'lar ile storage arasındaki bağlantı teknolojisi ne kadar hızlı ise veri aktarımı o kadar hızlı gerçekleşir. Bu da anlık okuma/yazma gerçekleştiren VM'ler için daha performanslı bir çalışma ortamı sağlar.

Bağlantı teknolojisi dışında storage üzerindeki disklerin tipi ve hızlarının da önemli olduğunu söylemiştik. SATA, SCSI, SAS ve 7200rpm, 10000rpm, 15000rpm gibi disk tipleri de performansa etki edecektir.

Ayrıca disklerin çalıştığı kontroller kartlar, bu kartların memory durumları ve disklerin raid yapılandırılmaları da performans için belirleyici olmaktadır.

Bu noktada daha önce de bahsettiğimiz maliyetler devreye giriyor çünkü sağlam bir Fibe Channel yapısı kurmak gerçekten ciddi rakamlara mal oluyor.

Biz bu makalemizde yazılımsal bir iSCSI storage kullanarak daha düşük maliyetli failover cluster senaryolarını nasıl oluşturabiliriz ve kurduğumuz bu yapıyı nasıl maksimum performans ile kullanabiliriz bakacağız. Ayrıca bu makale size Hyper-V'teki failover cluster mantığını öğretecek ve aynı konfigürasyonları fiber channel storage'ler ile de gerçekleştirebileceksiniz.

Tekrar hatırlatmakta fayda görüyorum. Özellikle yüksek performans ihtiyacı olan sanallaştırma ortamlarda ve mission-critical rollerin barındığı sunucularda Fiber Channel Storageler kullanmak performans ve güvenilirlik açısından daha doğru bir karar olacaktır. Bu makalede ele alacağımız software tabanlı iSCSI storage'leri production ortamlarda kullanırken dikkatli olmanızı, testlerini iyi yaparak beklentilerinizi karşılayıp karşılamadığını analiz etmenizi ve kesinlikle best practice yöntemlerine uymanızı öneriyorum.

iSCSI arabirimine sahip bir storage, appliance yani donanımsal ürünler olabileceği gibi software yani yazılımsal ürünler de olabilir. Belki software de bir donanım üzerinde çalışıyor ama burada takılmamız gereken nokta bu değil ☺

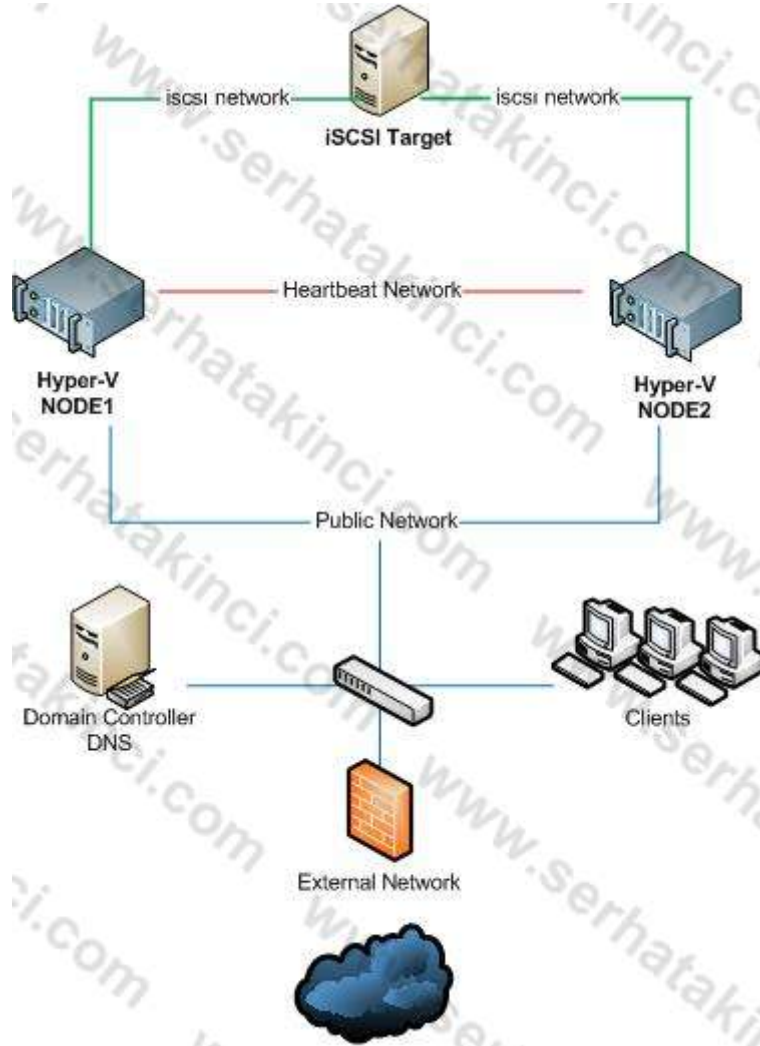
Software temelli storageler genelde bir işletim sistemi üzerine kurulan tool'lar sayesinde çalışır ve zemindeki işletim sistemine bağımlıdır. İşletim sisteminin çalıştığı server üzerindeki fiziksel diskleri ve ya bu diskler üzerinde yaratılan birimleri iscsi protokolü ile uzak kullanıcıların hizmetine sunarlar.

Örneğin Windows Storage Server, üzerindeki **iSCSI Software Target** sayesinde local disk üzerindeki birimleri (ör: VHD) network üzerindeki sistemlerin kullanımına sunabilmektedir. Openfiler gibi opensource uygulamalarda aynı işi rahatlıkla yapabilirler.

Amacımız Hyper-V'teki failover cluster yapısını incelemek ve düşük maliyetli bir çözüm üretmek olduğu için biz bu makalede iSCSI Storage olarak **Windows Server 2003 Standard Ed.** Üzerinde çalışan **Microsoft iSCSI Software Target** yazılımını kullanacağız.

#### 4. Örnek Topoloji ve Temel Gereksinimler

iSCSI Software Storage kullanacağımız failover cluster yapısı için oluşturacağımız topology'i ve server'ların konumunu görelim.



Diagramı inceledikten sonra Topology'i biraz açalım ve nelere ihtiyacımız olduğuna bakalım.

Gördüğümüz gibi arka planda çalışan bir storage var (iSCSI Software Target) ve söylediğimiz gibi bu hizmeti **Windows Server 2003 STD** üzerinde çalışan **Microsoft iSCSI Software Target** yazılımı veriyor. VM'ler bu storage üzerinde duruyor olacak.

VM'lerin çalışacağı iki fiziksel Node var. Hyper-V Node1 ve Hyper-V Node2. Bu iki Node üzerinde **Windows Server 2008 Enterprise Edition 64bit** işletim sistemi çalışıyor. Biliyorsunuz sadece Enterprise ve Datacenter sürümler Cluster destekler (Ayrı HPC).

Her iki node üzerinde 3'er adet fiziksel NIC var. Doğru ve performanslı bir iSCSI cluster yapısı için bu NIC'ler gerekli.

**1nci NIC:** Public network'e bakıyor yani kullanıcıların ve diğer server'ların birbiri ile konuştuğu, domain iletişiminin gerçekleştiği network.

**2nci NIC:** Heartbeat network'e bakıyor. Bu network, cluster üyesi iki Node'un birbirlerini yokladığı, ayakta olup olmadıklarını (up) kontrol ettikleri network olacak ve diğer networklerden bağımsız olacak.

**3ncü NIC:** iSCSI network'e bakıyor. Bu network üzerinde sadece ve sadece iSCSI trafiği akacak ve birer NIC bu işi için dedicate edilmiş olacak.

Ayrıca Node'lar Hyper-V çalıştıracak :) Yani **64bit** ve **hardware virtualization** destekli processor'ler şart.

Yapımızda bir Domain Controller yani active directory domain'i ve DNS olmak zorunda. Cluster için domain ortamı şart.

Topology deki ihtiyaçları maddeler halinde toparlarsak:

#### **Storage:**

- Windows Server 2003 STD Ed.
- Microsoft iSCSI Software Target
- En az iki yada daha fazla sayıda NIC (1gbps tavsiye edilir)
- İhtiyaç kadar fiziksel disk alanı
- Domain üyeliği

#### **Hyper-V Node'lar:**

- Hyper-V uyumlu donanım ve processor
- Windows Server 2008 Enterprise Edition 64bit
- Hyper-V'nin RTM olmasını sağlayan KB950050 paketi yüklenmiş olmalı (Windows server 2008 SP1 için).
- SP2 önerilir. (Her iki node üzerindeki SP durumu aynı olmalı)
- En az 3 adet fiziksel NIC (performans için tavsiye edilir)
- Domain üyeliği
- iSCSI initiator (Win2008 içerisinde geliyor)
- Storage üzerinden atanmış disk birimleri (Quorum ve VM'ler için)

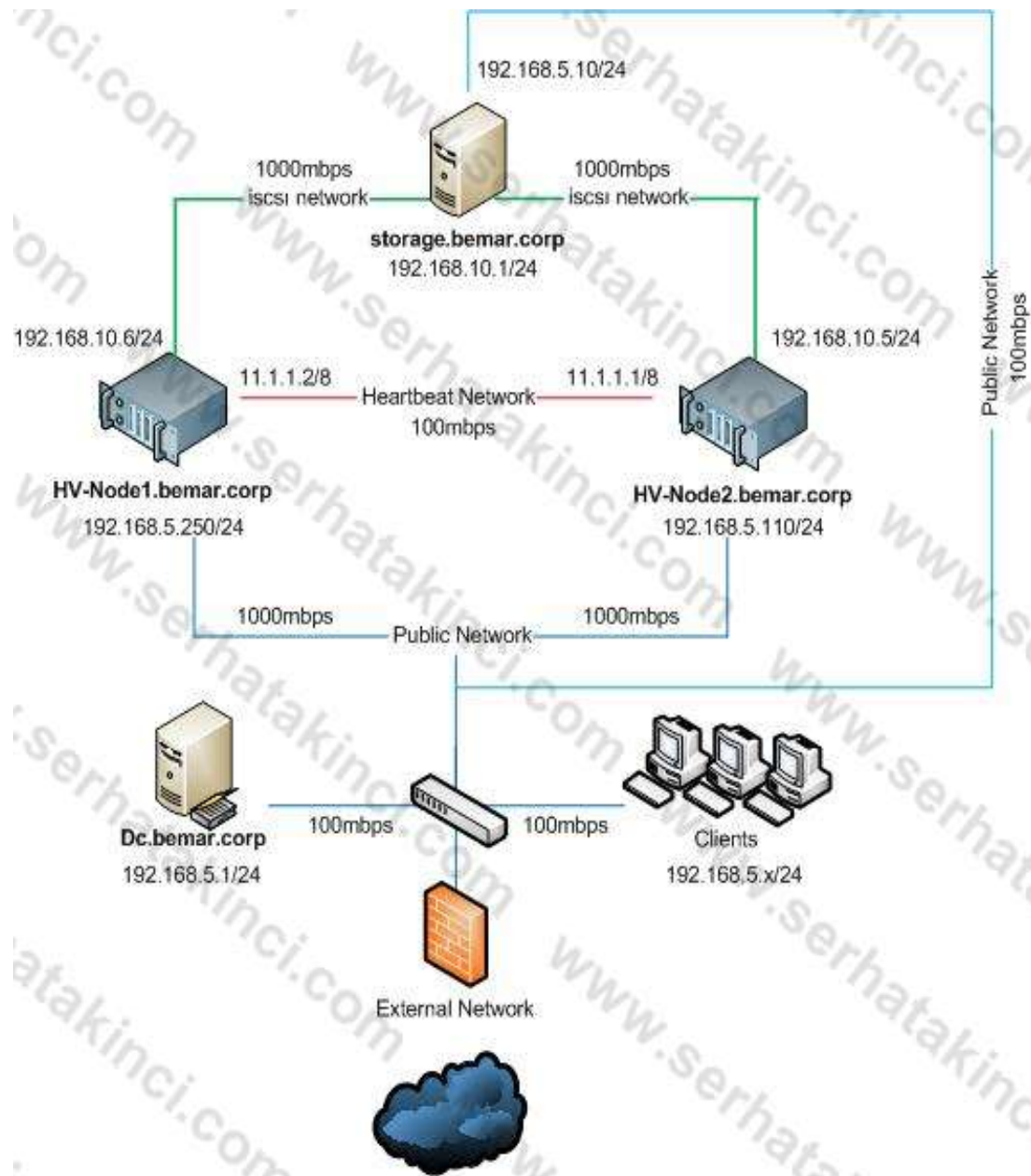
#### **Domain Controller:**

- Windows Server 2008 (2003te olabilir)
- AD Domain Services ve DNS rolleri

#### **Network:**

- 1nci network: 192.168.5.x – 255.255.255.0 (public network)
- 2nci network: 11.1.1.x – 255.0.0.0 (heartbeat network)
- 3ncü network: 192.168.10.x – 255.255.255.0 (iSCSI network)

Topology'nin domain ortamına oturtulmuş ve ip'lendirilmiş hali aşağıdaki gibidir. Ayrıntıya girdiğim için ilk bakışta karışık görünebilir ama dikkatli incellerseniz kolayca anlayabildiğinizi göreceksiniz.



## 5. iSCSI Storage Konfigürasyonu:

Storage yapılandırması ile başlayalım.

Gereksinimlerde de belirttiğimiz gibi storage görevi görece makine üzerinde **Windows Server 2003 STD 32bit** işletim sistemi çalışıyor. Bu işletim sisteminin kurulumu konusuna girmiyorum çünkü normal kurulum aşamaları dışında özel bir durum yok. İşletim sisteminin kurulmuş, tüm update'lerin yapılmış olması yeterli.

### 5.1. IP ve Domain Ayarları

Üzerinde en az iki NIC'e ihtiyacımız olduğunu söylemiştik.

İlk NIC storage makinesinin domain ve public iletişimi için **public network**'e bakmalı ve 192.168.5.x/24 aralığından ip adresi olmalı.

İkinci NIC'i ise iSCSI iletişimi için kullanacağız ve **iSCSI network**'e bakacak. 192.168.10.x/24 aralığından ip adresi olacak.

Bu iki network'ü birbirinden ayırmamızdaki amaç şu: iSCSI veri paketlerinin network yani ethernet üzerinden iletildiğini söylemiştik. Eğer ki iSCSI paketlerini public network üzerinden gönderirsek, o network üzerindeki diğer sistemlerin yaratmış olduğu trafik iSCSI paketlerini olumsuz etkileyebilir, daha da önemlisi zaten sınırlı olan veri aktarım hızını düşürebilir. Bu nedenle sadece bu iş için ayrılmış bir network ve network interface card'lar kullanmak performansı ciddi anlamda arttıracaktır çünkü ilgili network bandwidth tamamen bu işe dedicate edilmiş olur. Ayrıca iSCSI network'ü gbit network ekipmanları (nic, switch, cable) ile kurmanız şiddetle tavsiye edilir çünkü iletim hızı network bandwidth ile doğru orantılıdır.

Benim senaryomdaki storage için ip config. şu şekilde.

#### Public interface (NIC)

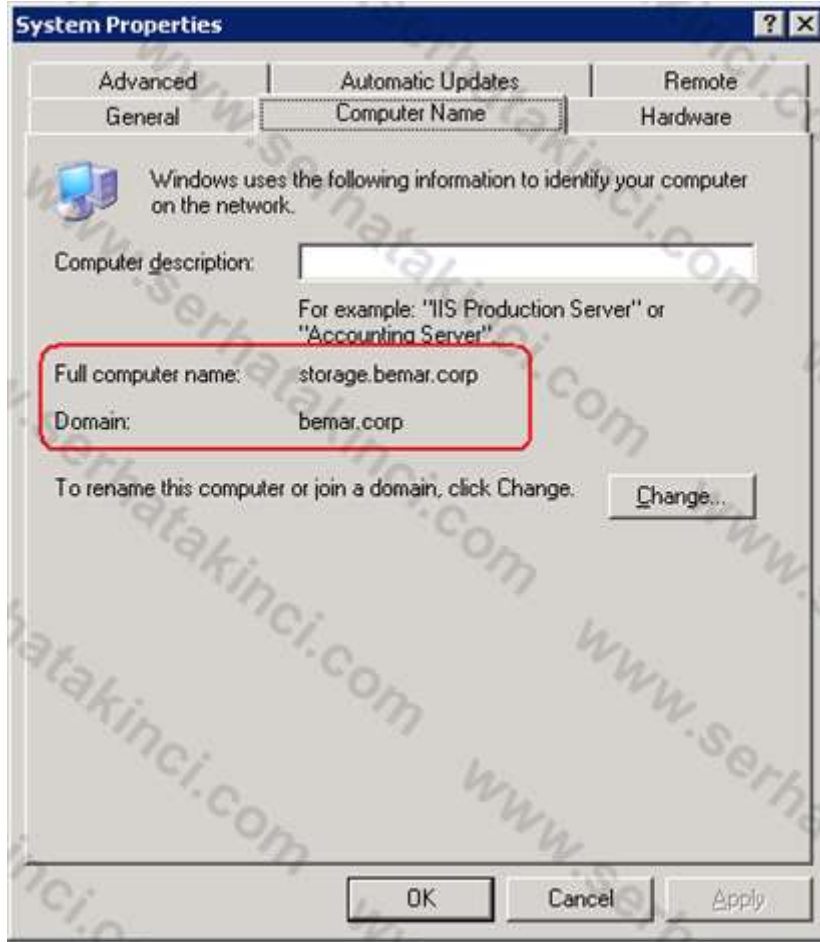
IP : 192.168.5.10  
Mask : 255.255.255.0  
DNS : 192.168.5.1

#### iSCSI interface (NIC)

IP : 192.168.10.1  
Mask : 255.255.255.0

Gerekli ip config. yaptıktan ve NIC'leri uygun fiziksel switch'lere bağladıktan sonra Windows Server 2003 çalıştıran storage makinesini domain'e üye yapıyoruz.





Ayrıca storage üzerindeki Windows firewall açıksa ya da third-party bir güvenlik yazılımı çalışıyor ise **3260** numaralı porta izin vermemiz gerekir çünkü iSCSI iletişimi bu port üzerinden olacak.

Daha sonra üzerine **Microsoft iSCSI Software Target** kurulumu yapıyoruz.

## 5.2. Microsoft iSCSI Software Target Kurulumu

**Microsoft iSCSI Software Target** OEM kanalından temin edilebilen bir ürün. Ya da **Application Pack** ile **Windows Storage Server** üzerinde kullanabilirsiniz. Ayrıca **Windows Unified Data Storage Server** içerisinde de yerleşik olarak geliyor.

Bu ürün hakkında bazı soru ve cevaplara aşağıdaki link ile ulaşmanız mümkün.

<http://www.microsoft.com/windowsserversystem/storage/iscsifaq.mspx>

Eğer Microsoft iSCSI Software Target temin edemiyorsanız herhangi bir iSCSI Software Target kullanabilirsiniz. **Openfiler** yada **StarWind** gibi.

Microsoft iSCSI Software Target kurulumu oldukça basit. Hatta sıradan bir update paketini yüklemekten farkı yok.

Hızlıca kurulumu yapalım.

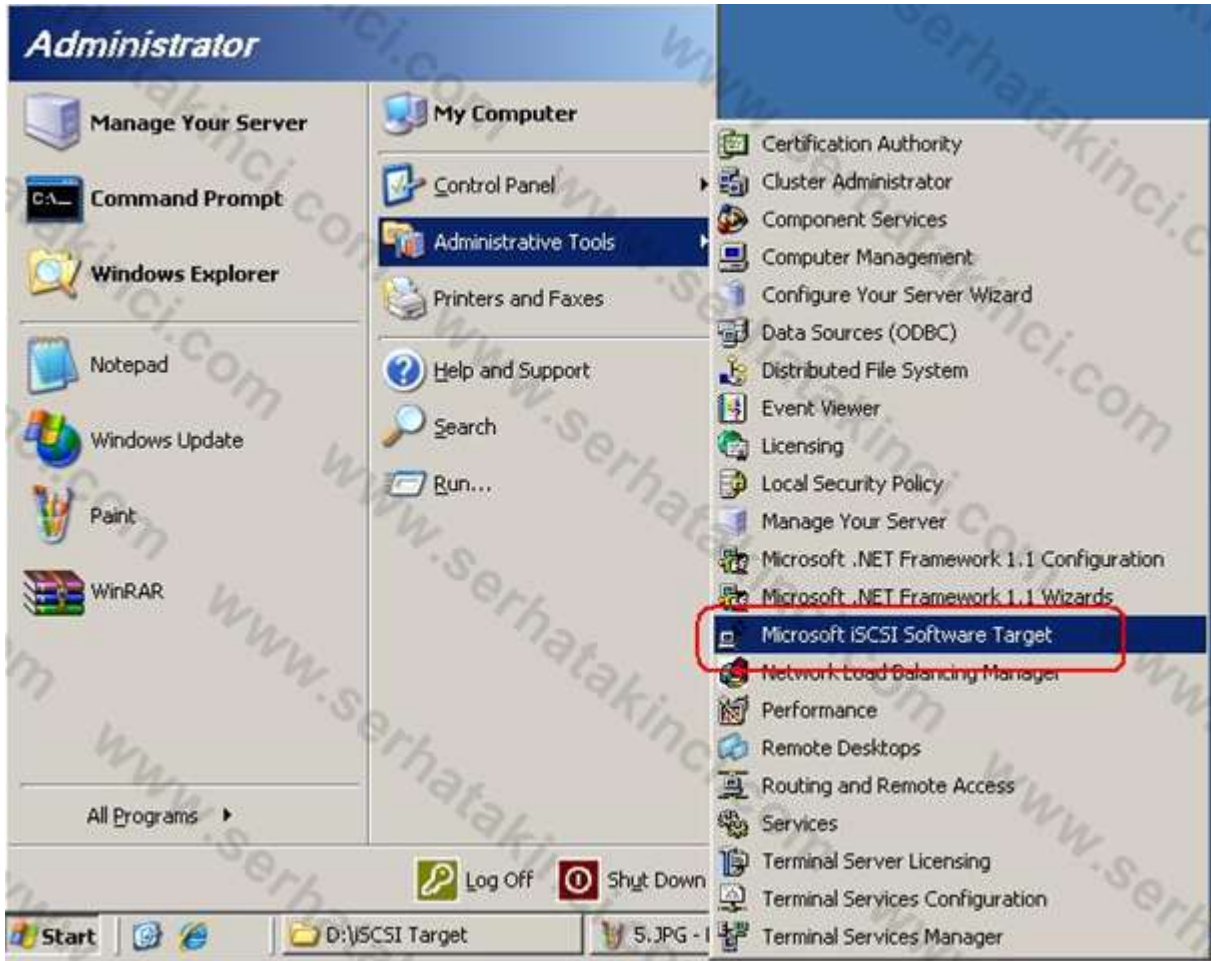
Kurulum için **iscsitarget-x86.exe** çalıştırıyorum.



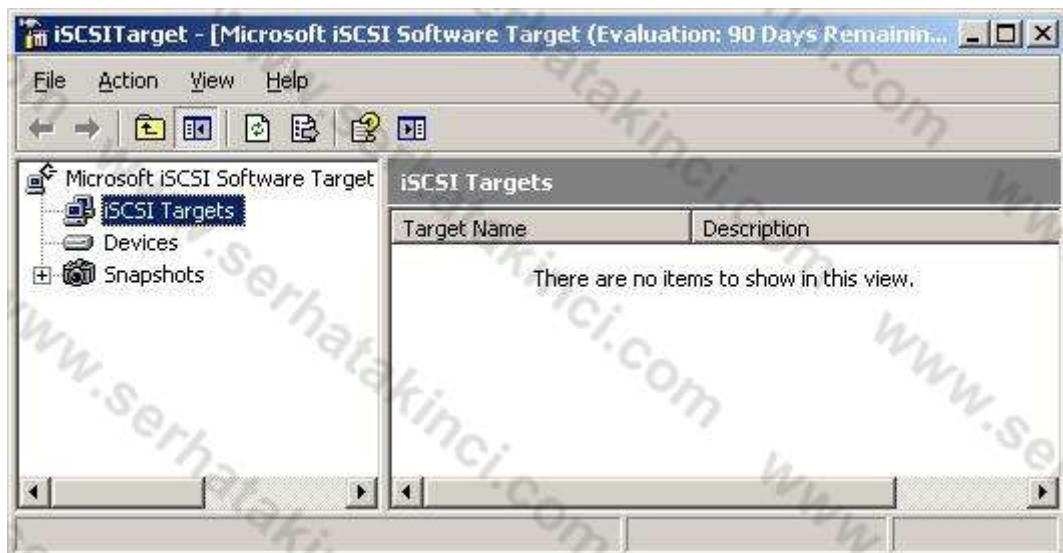




Gördüğünüz gibi kurulum tamamlandı. Bu kurulumdan sonra **Microsoft iSCSI Software Target Management** konsol, **administrative tools** altındaki yerini almış olur.



Konsolu açtığınızda basit birkaç bölümden oluştuğunu göreceksiniz.



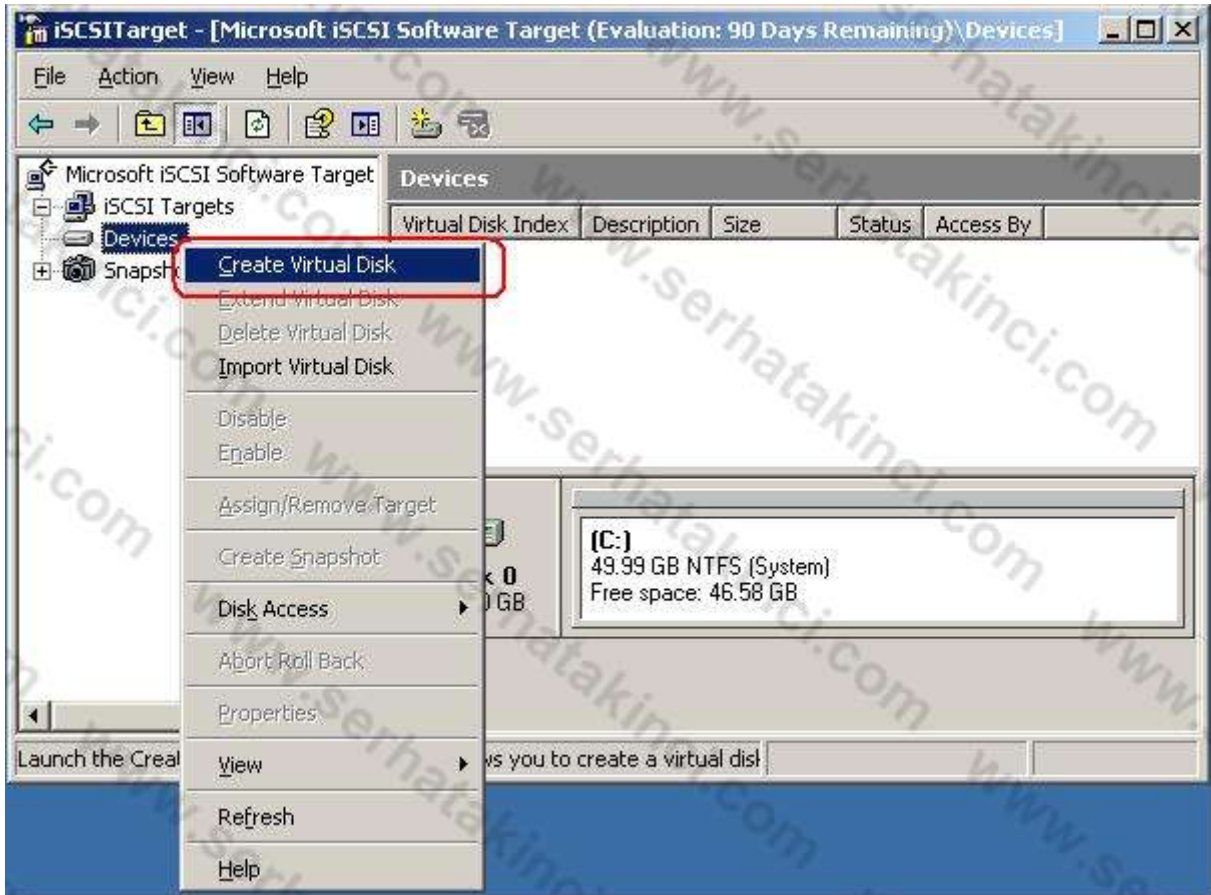
### 5.3. Microsoft iSCSI Software Target Üzerinde Virtual Disk Yaratmak

Şimdi storage üzerinde depolama birimleri yaratalım. Bu birimler Microsoft iSCSI Software Target üzerinde **Virtual Disk** olarak adlandırılır. Bu diskleri daha sonra Hyper-V Node'larına atayacağız ve cluster için kullanacağız.

İlk etapta iki disk yaratacağız. İlk **VM** için. İkincisi ise **Quorum** yani cluster bilgisinin tutulacağı bölüm için.

İlk disk 30GB olarak yaratalım.

**Devices** üzerinde **Create Virtual Disk** diyoruz.



**Virtual Disk Wizard** geliyor.



Oluşacak disk için path ve isim belirtiyoruz.

Dikkat ederseniz disk **VHD** formatında. Yani storage makinesi üzerinde yaratılan VHD formatlı sanal diskleri, iSCSI üzerinden Hyper-V Node'lara tahsis etmiş olacağız.





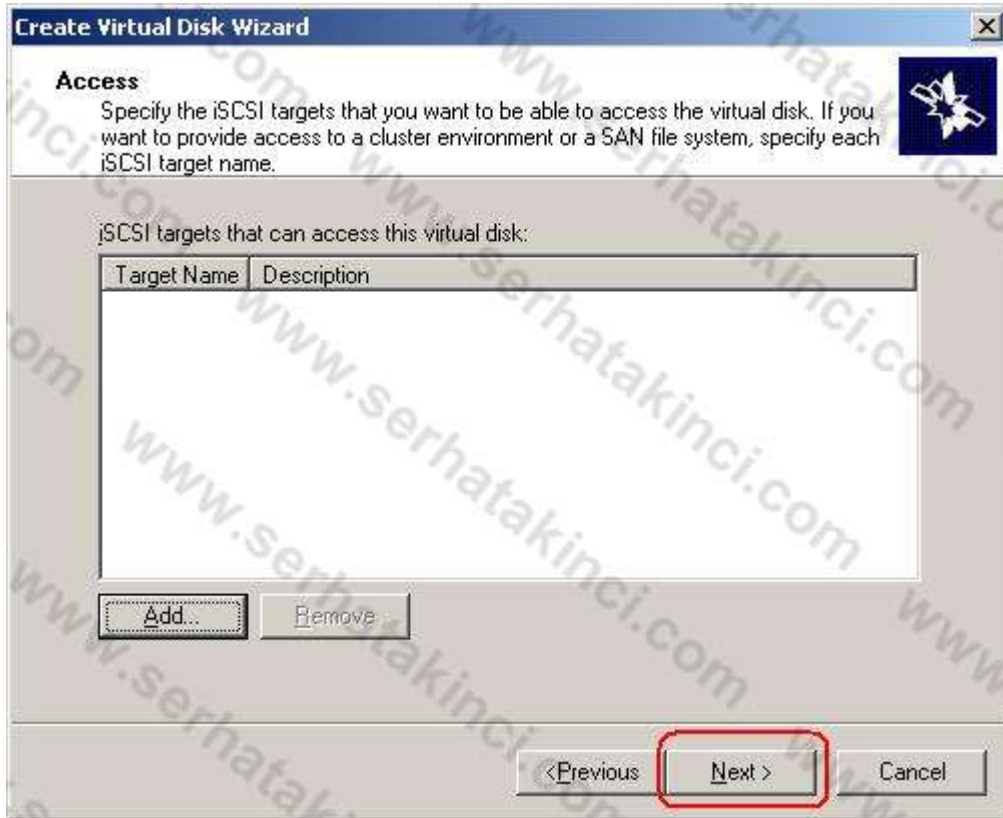
İlk Virtual Disk için **MB** cinsinden **size** belirliyoruz. 30GB karşılığı olan 30720MB veriyorum.



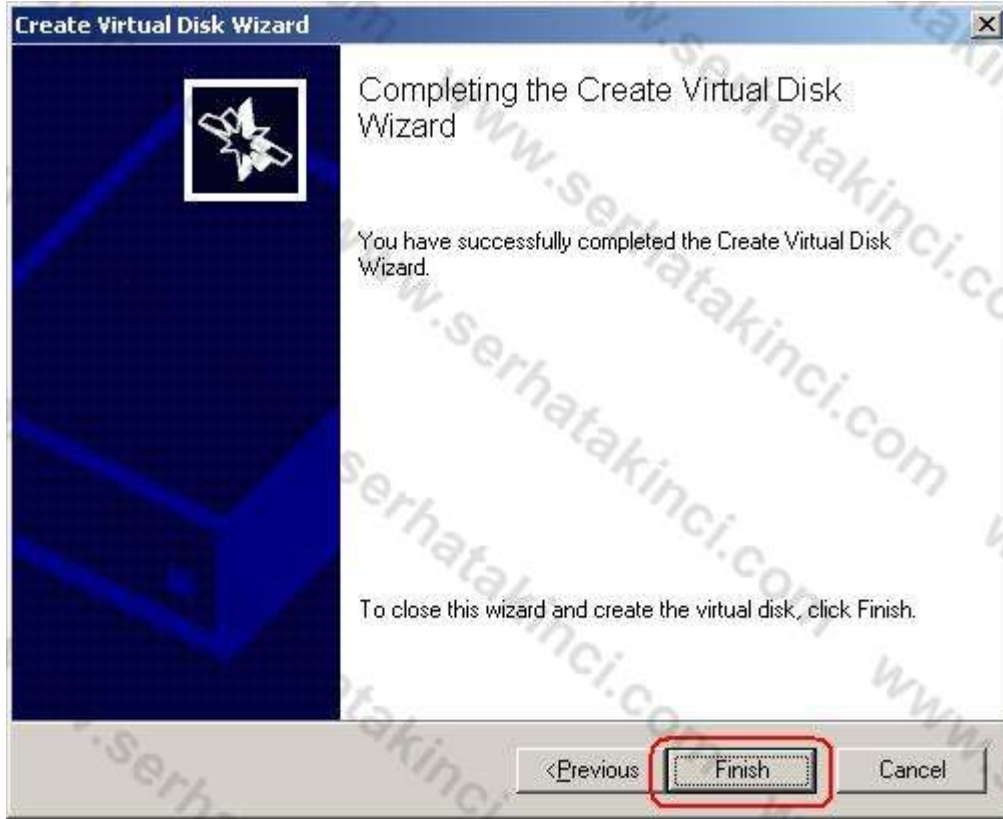
Yaratılacak Virtual Disk için bir açıklama girebilirsiniz. Opsiyoneldir.



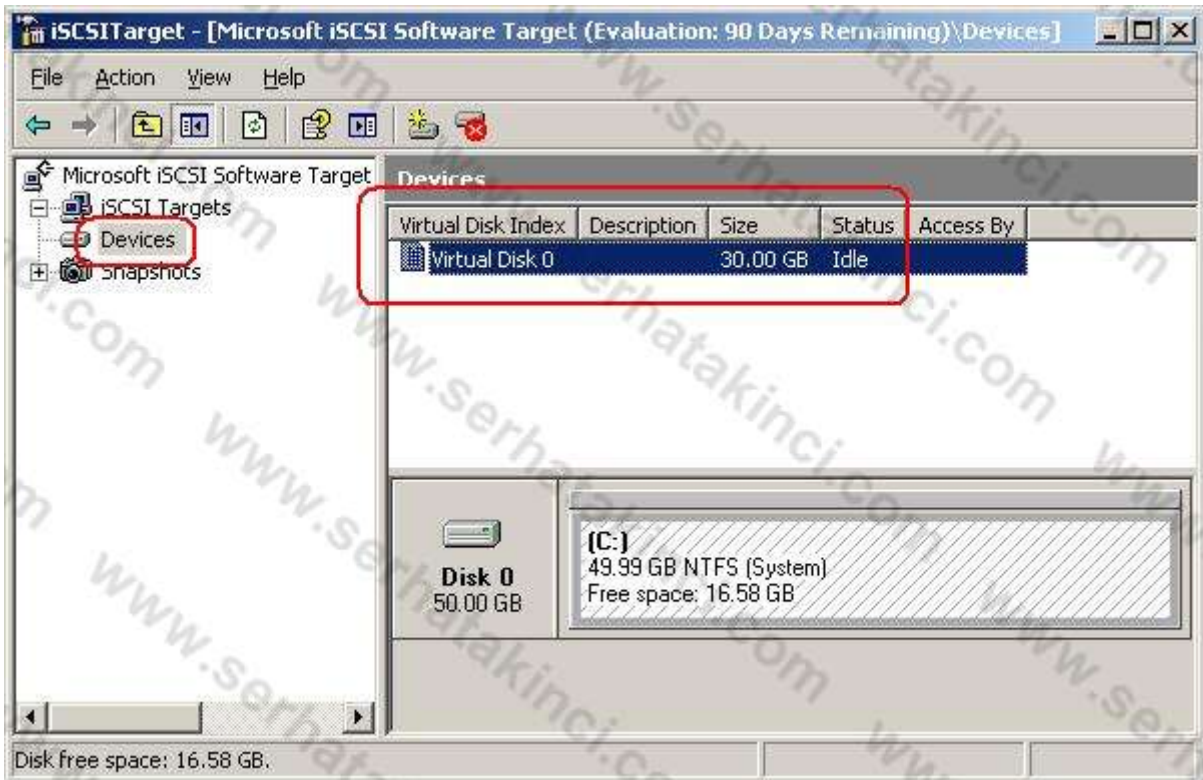
Aşağıdaki pencerede yaratılacak Virtual Disk'e kimlerin erişeceğini belirleyebiliyoruz (iscsi targets) ancak bunu daha sonra yapacağız çünkü henüz storage'e erişen node yok. Bu nedenle şimdilik boş geçiyoruz.



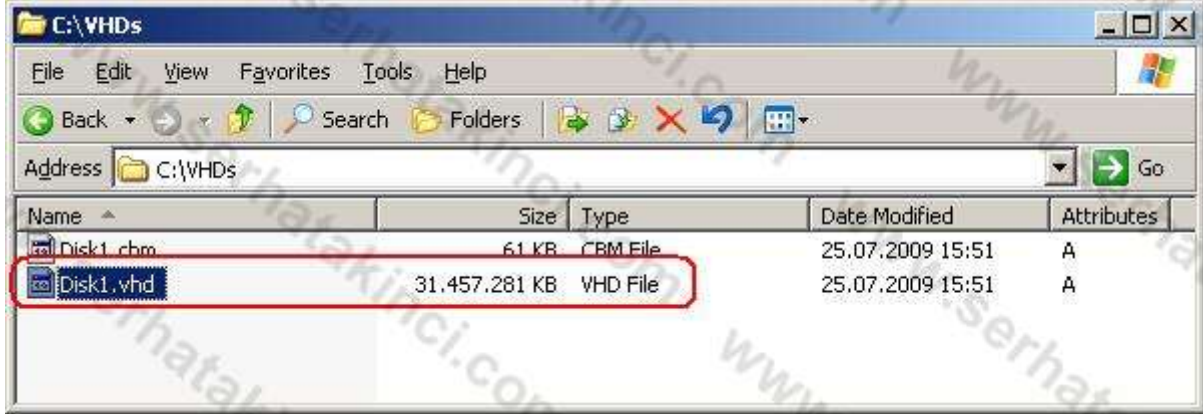
**Finish** dedikten sonra ilgili **Virtual Disk** yaratılır.



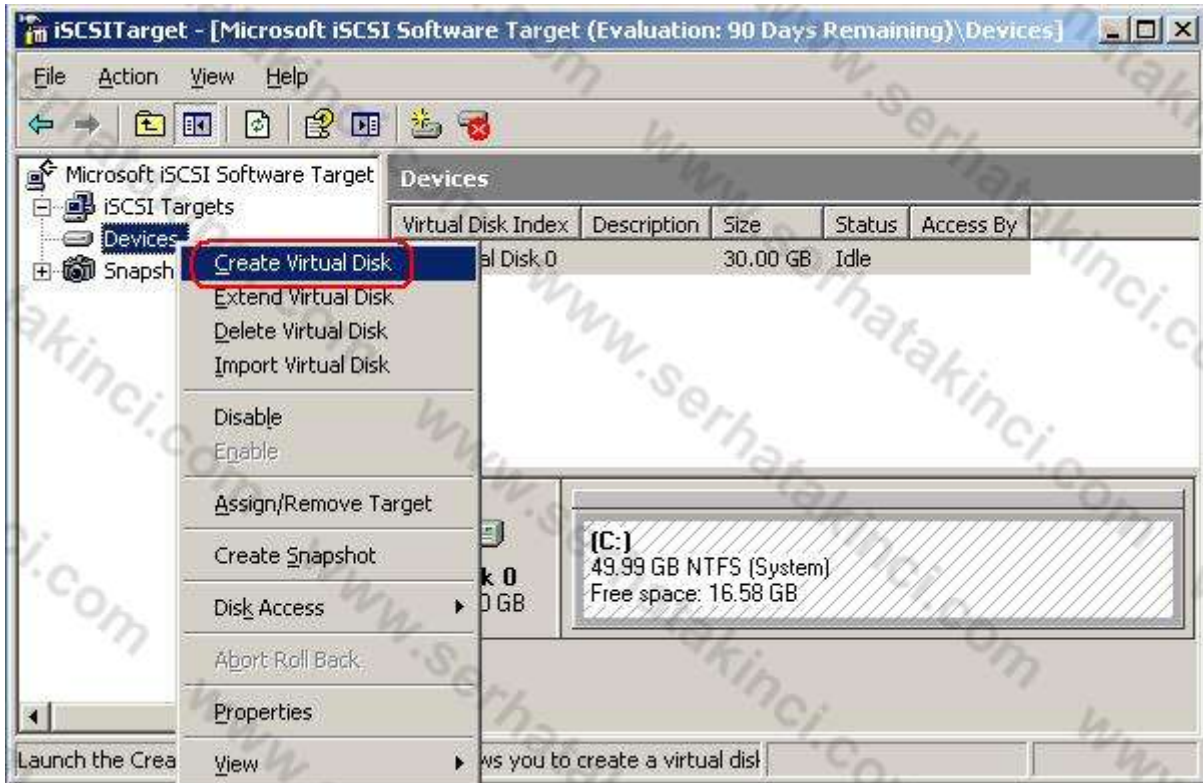
Konsol üzerinde oluşan **Virtual Disk**'i, **size**'ı ve **status**'u görebilirsiniz. Durumu **idle** olarak görünüyor çünkü bu disk henüz herhangi bir sunucuya atamadık.



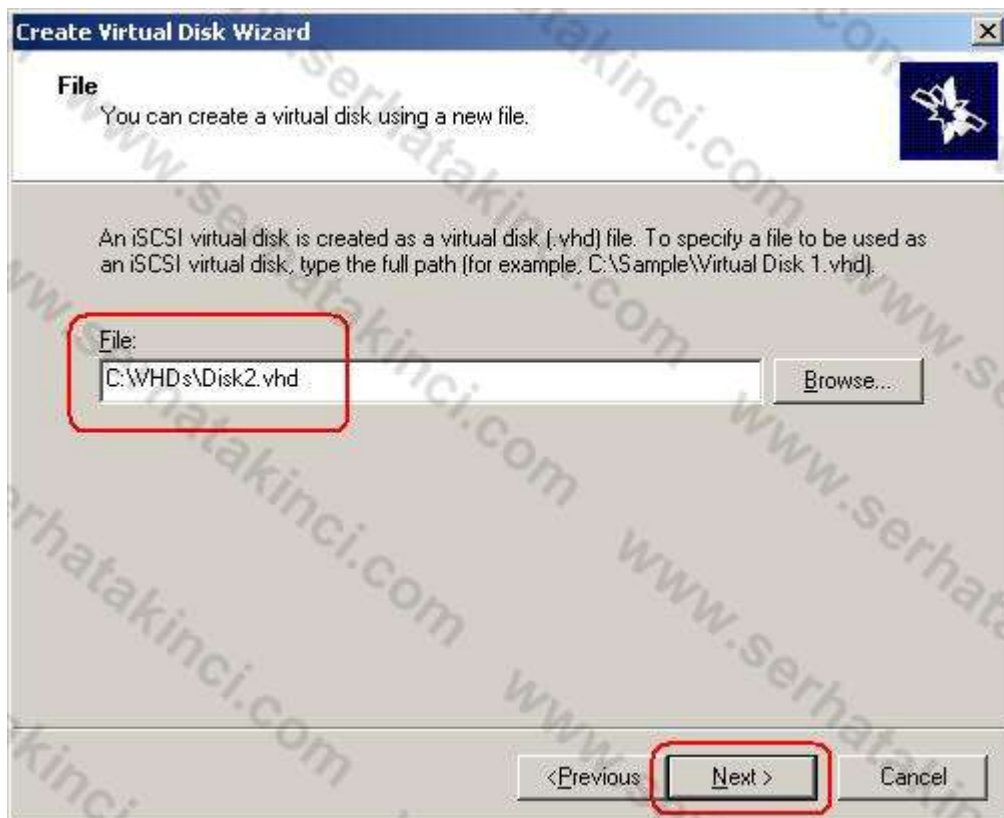
Virtual Disk için gösterdiğimiz path altında ise 30GB'lık bir VHD oluşmuş olmalı.



Şimdi aynı işlemi birde **Quorum** olarak kullanacağımız Virtual Disk için yapalım. Bu diskin boyutu ise 2GB olsun çünkü şimdilik Quorum için bu alan gayet yeterli.







**Create Virtual Disk Wizard**

**Size**  
Specify how much space on the volume to use for the virtual disk.

Current drive capacity: 49.99 GB  
Currently available free space: 16.58 GB  
Size of virtual disk (MB):

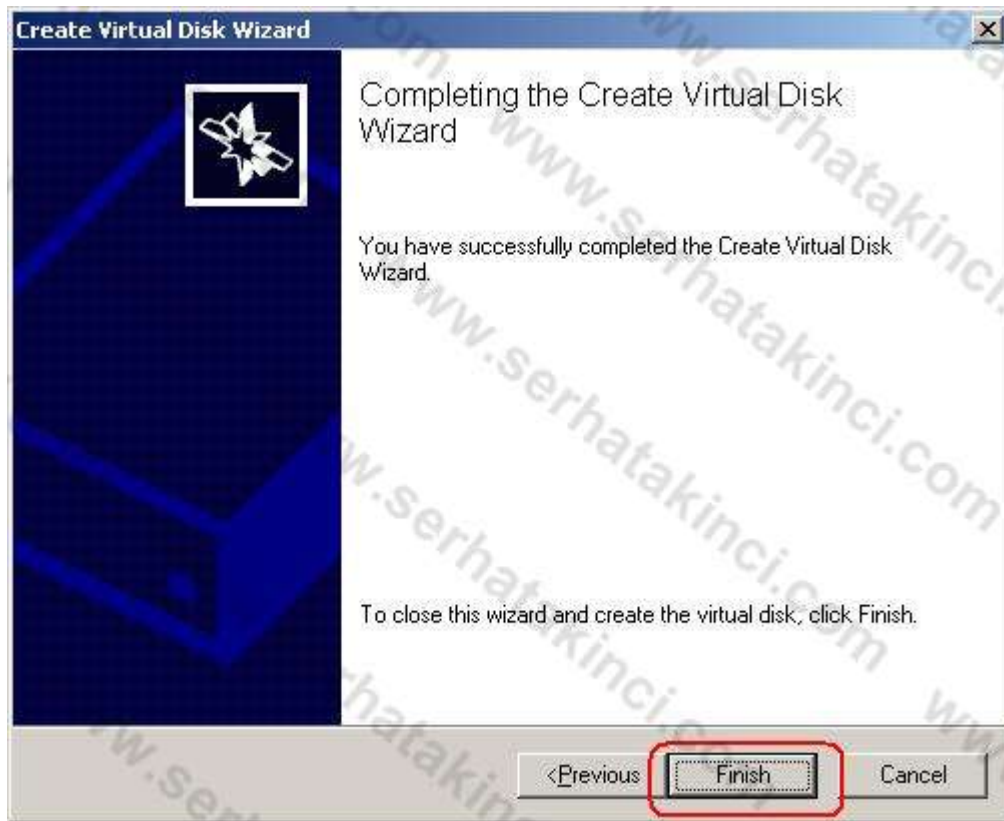
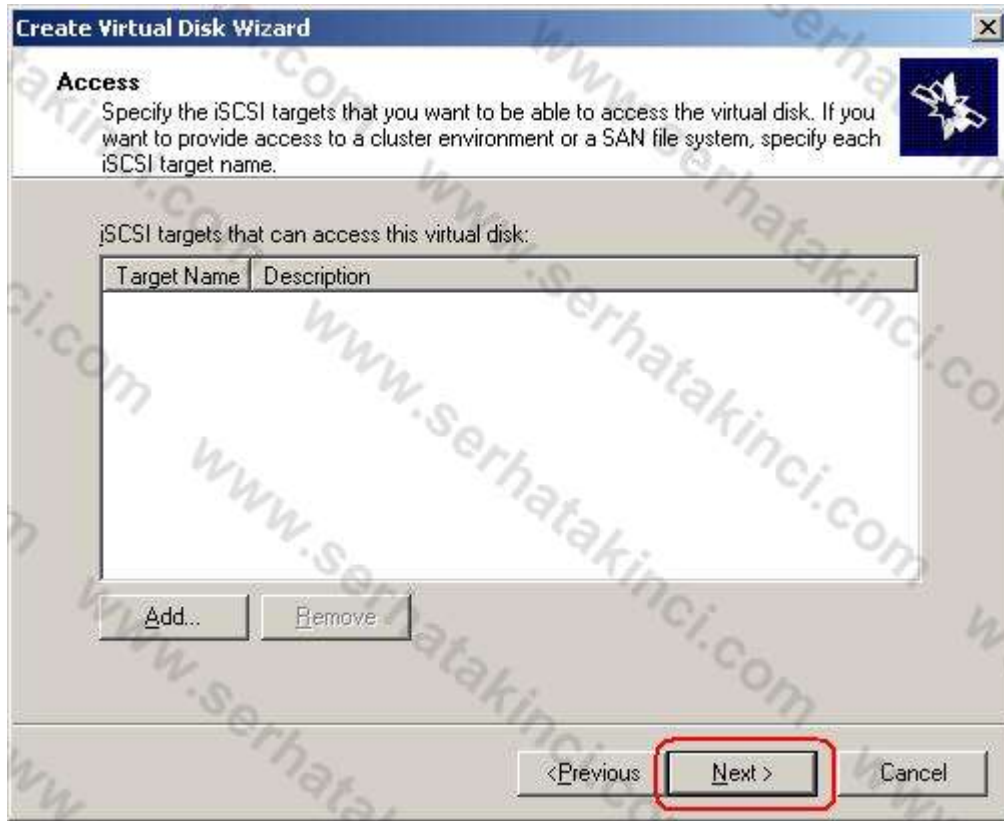
<Previous **Next >** Cancel

**Create Virtual Disk Wizard**

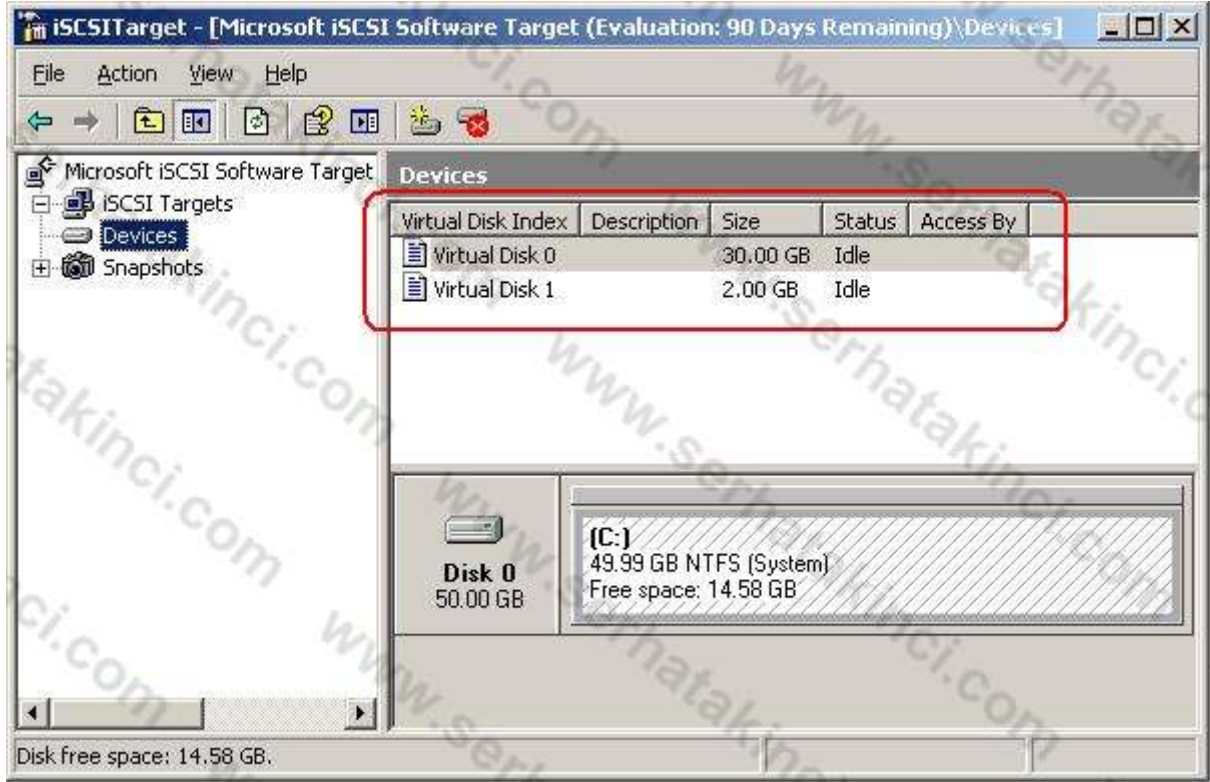
**Description**  
A description helps identify the virtual disk.

Virtual disk description:

<Previous **Next >** Cancel



İkinci virtual disk'i de yarattıktan sonra konsoldaki son durum aşağıdaki gibi olmalı.



Virtual Diskleri yarattık. Storage tarafındaki işimiz şimdilik bu kadar. Daha sonra diskleri atamak için tekrar storage makinesine döneceğiz.

## 6. Cluster Nodeların Konfigürasyonu

### 6.1. Genel Ayarlar

Hyper-V Nodeların yapılandırması ile devam edelim.

Nodelar üzerinde **Windows Server 2008 Enterprise x64** işletim sistemlerinin kurulmuş ve tüm Windows update'lerin yapılmış olması gerekiyor. Her iki node için SP (service pack) durumları da aynı olmalı. Yani bir node üzerinde SP1, diğer node üzerinde SP2 olmamalı.

Bununla birlikte sistem'ler Hyper-V rolünü çalıştırmak için hazır olmalı (yani donanımsal ve yazılımsal gereksinimler sağlanmış olmalı). Bu konu için daha önce hazırladığım Hyper-V kurulum makalelerine bakabilirsiniz.

Hyper-V rolünü cluster öncesinde ve ya cluster sonrasında ekleyebilirsiniz. Tavsiyem cluster sonrasında eklemeniz. Biz senaryomuzda cluster kurulumunu yaptıktan sonra ekleyeceğiz.

İşletim sistemi kurulumu, SP, updates, Hyper-V gereksinimleri gibi konulara girmeden, bu gereksinimleri yerine getirdiğinizi kabul ederek devam ediyorum.

Nodelar üzerindeki fiziksel NIC'leri yapılandırarak devam edelim. Hatırlarsanız her node üzerinde 3 adet fiziksel NIC vardı.

Benim senaryomdaki NIC ip config.'ler şu şekilde.

#### **HV-Node1 için ip config:**

##### Public interface (NIC)

IP : 192.168.5.250

Mask : 255.255.255.0

DNS : 192.168.5.1

##### iSCSI interface (NIC)

IP : 192.168.10.6

Mask : 255.255.255.0

##### Heartbeat interface (NIC)

IP : 11.1.1.2

Mask : 255.0.0.0

#### **HV-Node2 için ip config:**

##### Public interface (NIC)

IP : 192.168.5.110

Mask : 255.255.255.0

DNS : 192.168.5.1



iSCSI interface (NIC)

IP : 192.168.10.5

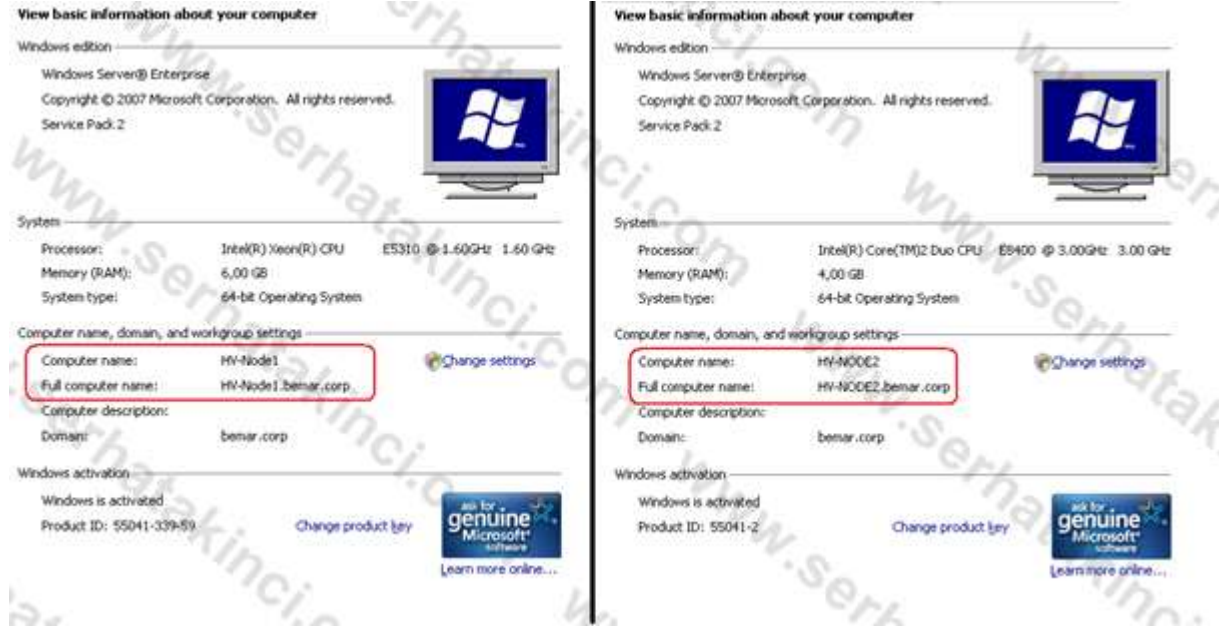
Mask : 255.255.255.0

Heartbeat interface (NIC)

IP : 11.1.1.1

Mask : 255.0.0.0

IP Config.'leri doğru şekilde ayarladıktan sonra tüm NIC'leri doğru fiziksel switch'lere bağlıyoruz ve her iki node'u domaine alıyoruz.



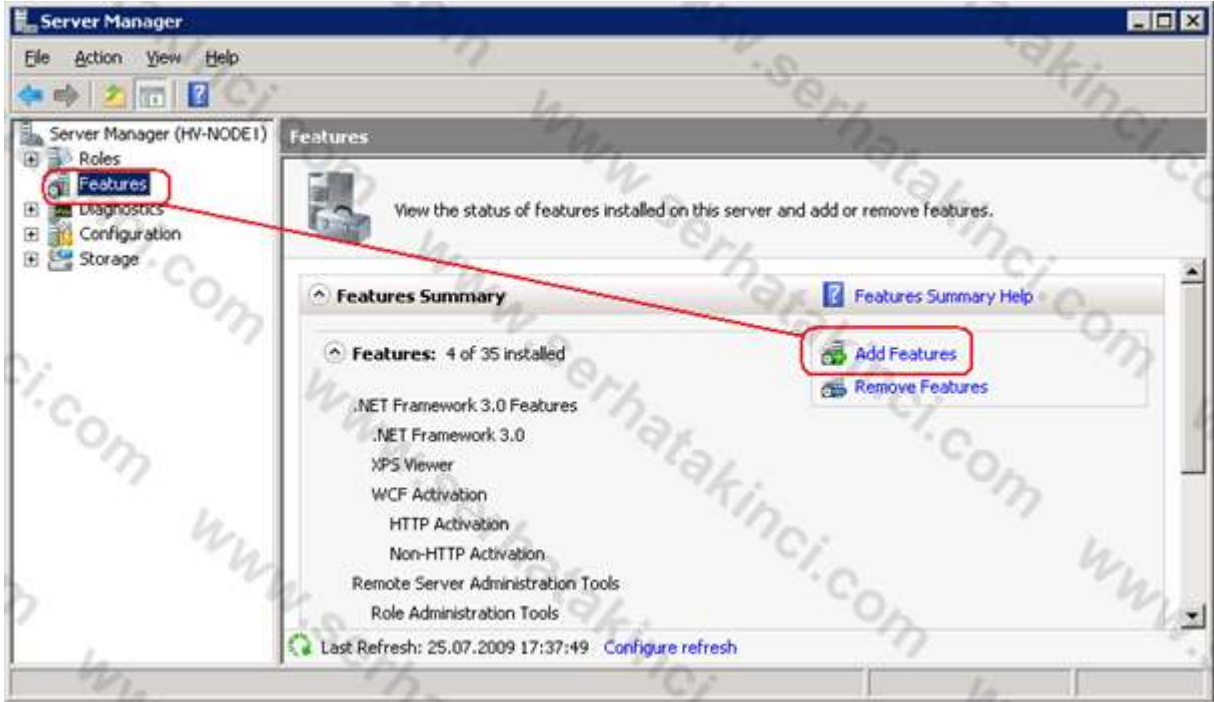
Makineleri domain'e aldıktan sonra artık cluster kurulumu için hazırız.

Öncelikle Cluster yapısı için gerekli ve her node üzerinde de olması gereken **Failover Clustering** ve **Multipath I/O** feature'ları yüklüyoruz (Multipath I/O feature's şart değil ancak yüklemenizi tavsiye ederim).

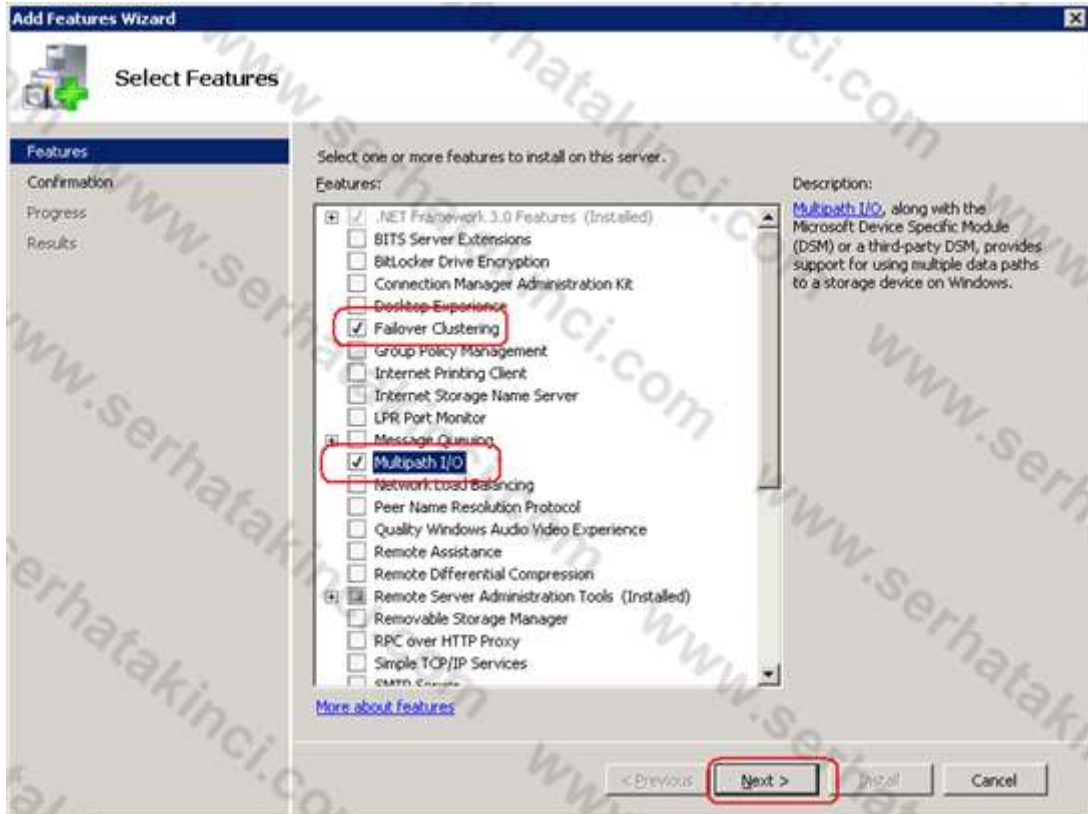
Yüklemeleri aşağıdaki gibi yapabilirsiniz.

## 6.2. HV-Node1 Üzerine Gerekli Feature'ların Yüklenmesi

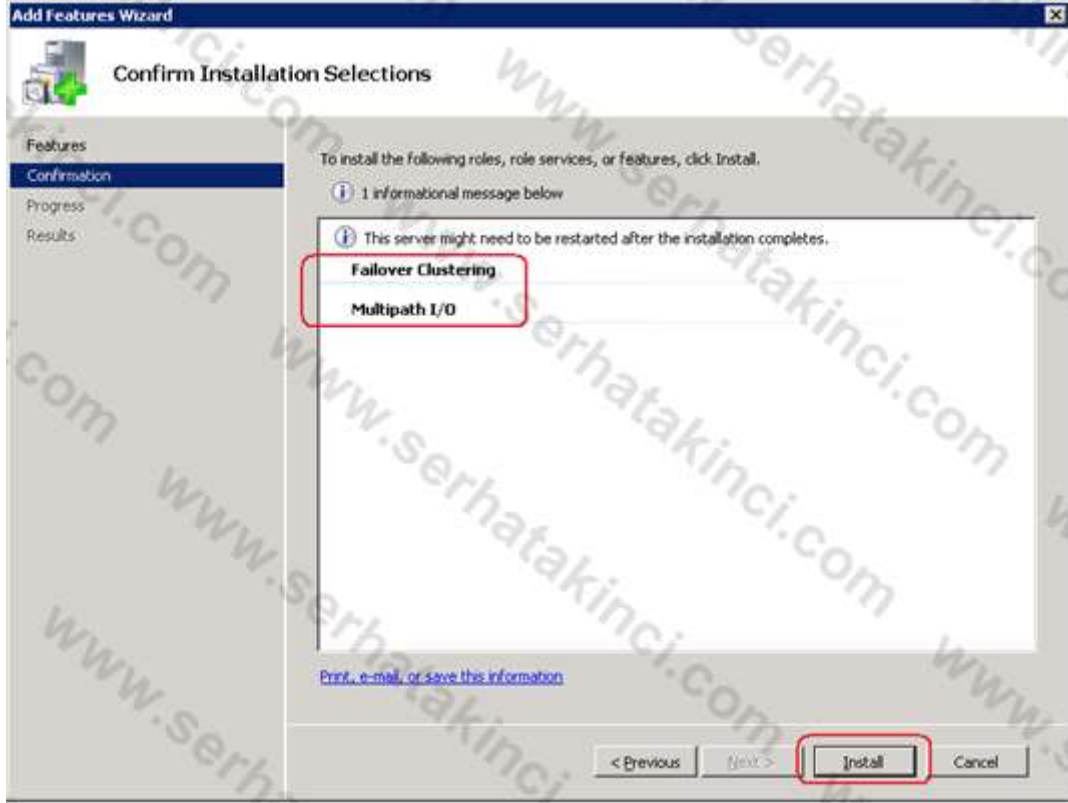
HV-Node1 üzerinde **Server Manager** konsolunu açıyoruz ve **features** bölümüne gelip **Add Features** diyoruz.



Gelen wizard ile iki feature'ı seçiyoruz. (**Failover Clustering** ve **Multipath I/O**)

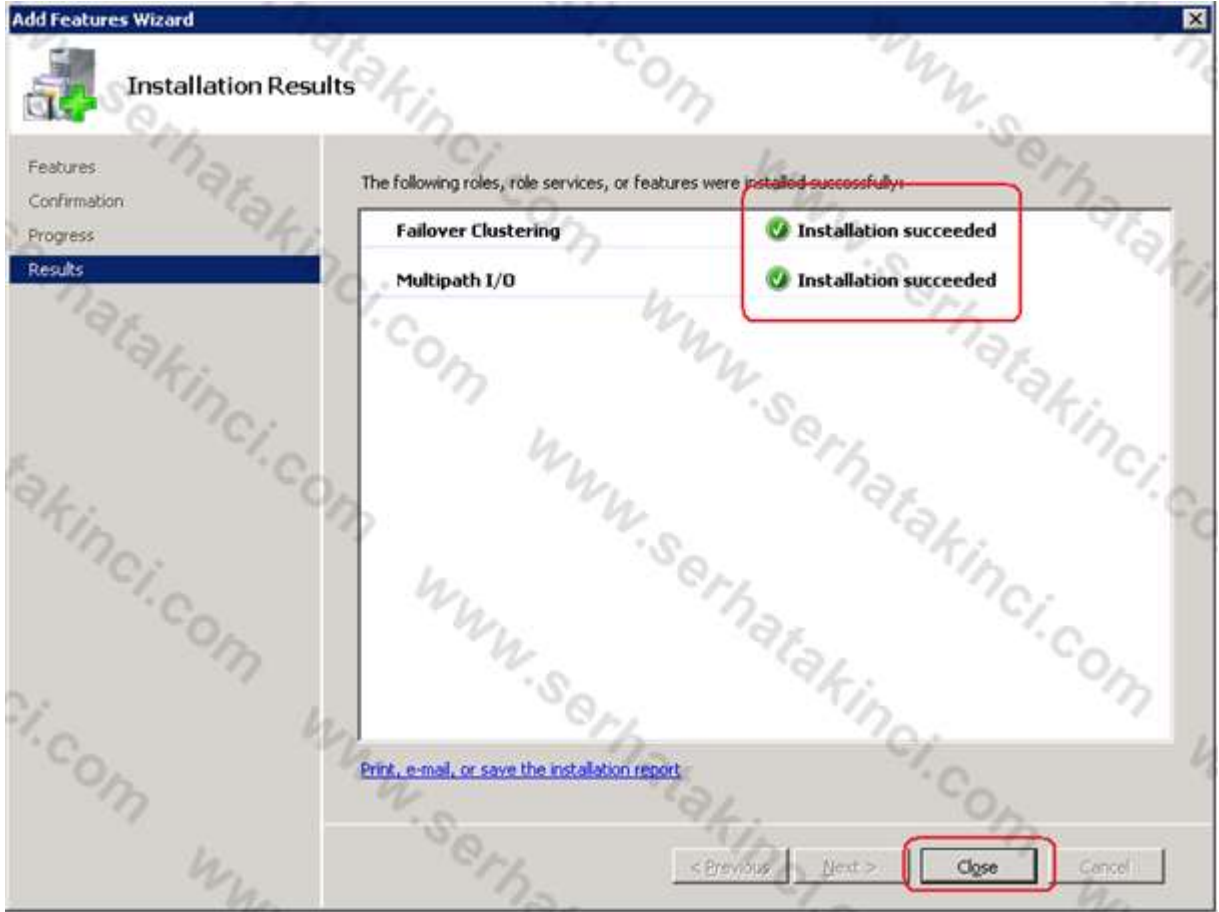


Ve **Install** diyerek kurulumu başlatıyoruz.

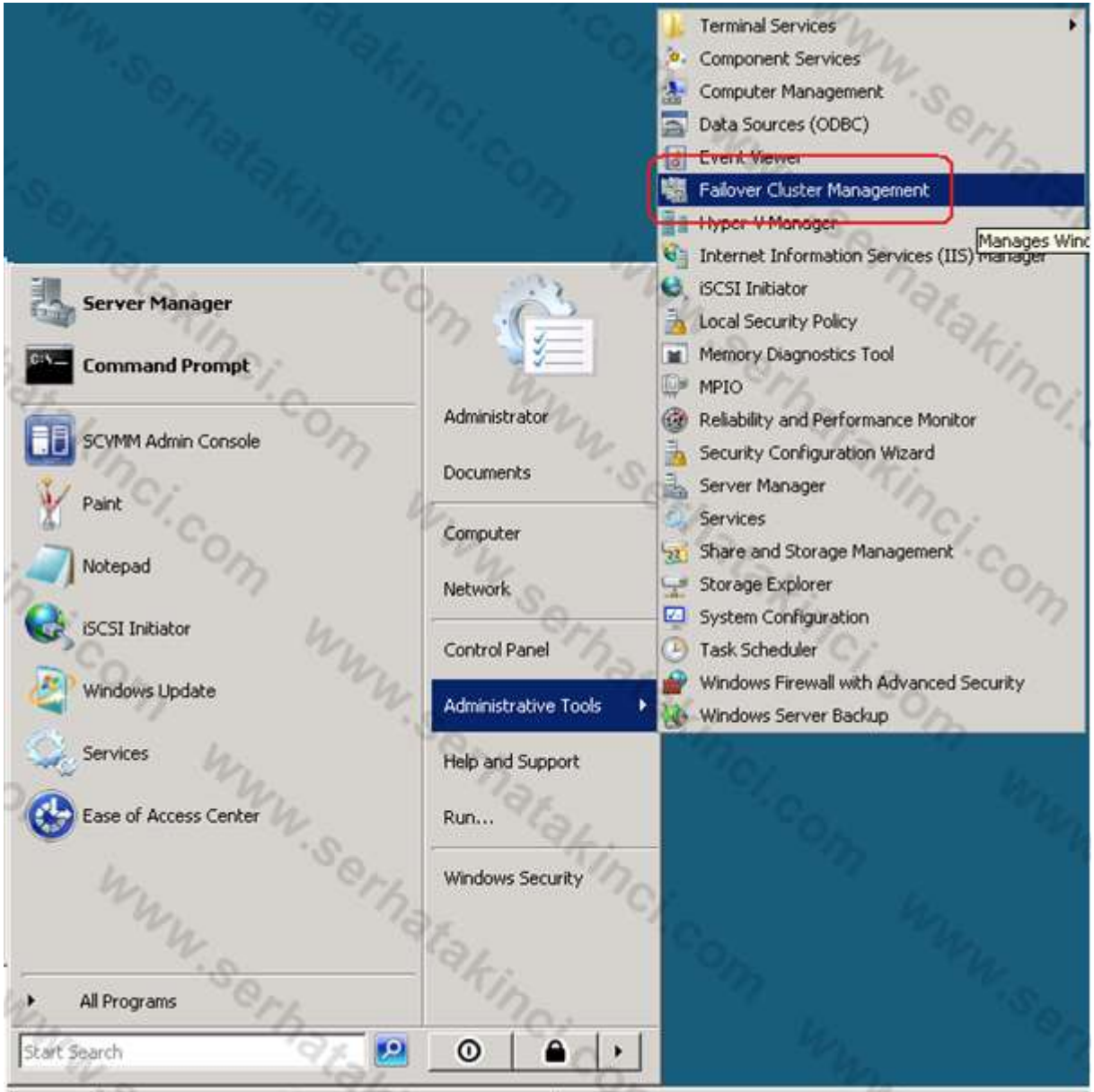


Kurulum tamamlandıktan sonra **succeeded** bilgisini görüyor olmalıyız. Aksi durumda sorun vardır ve kurulumu yeniden başlatmanız gerekebilir.





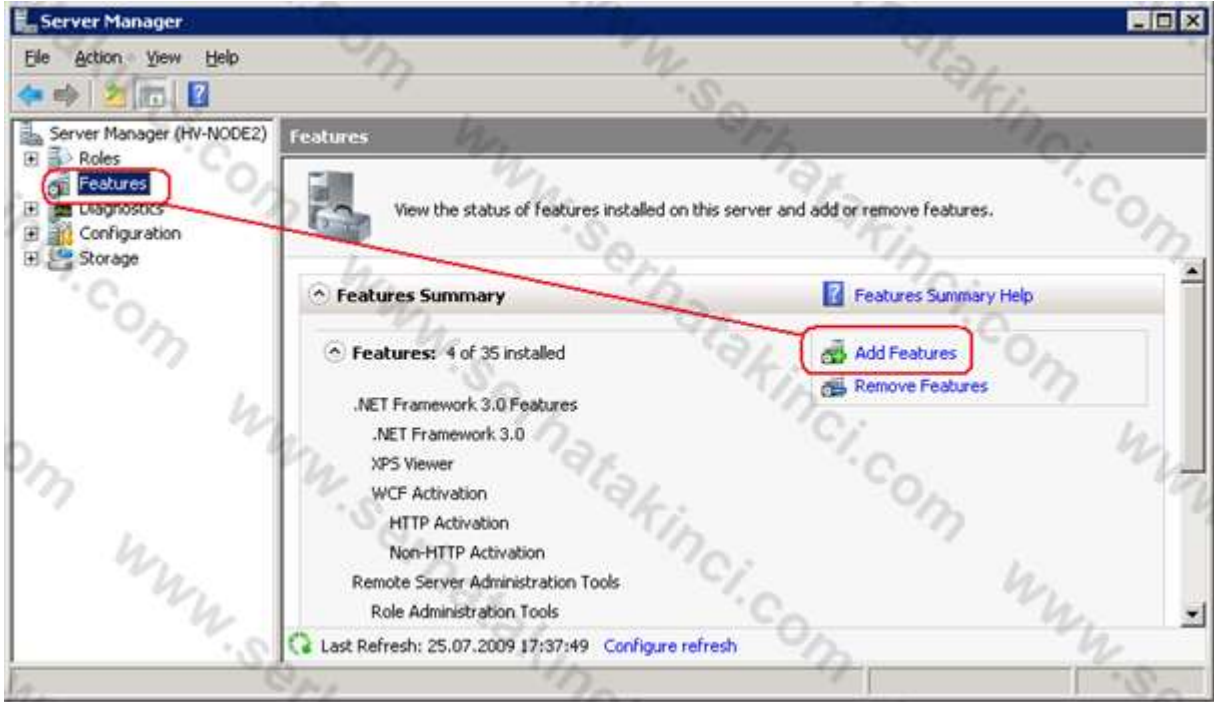
Yükleme işlemi tamamlandıktan sonra **Failover Cluster Management** konsolu **Administrative Tools** altındaki yerini alır.



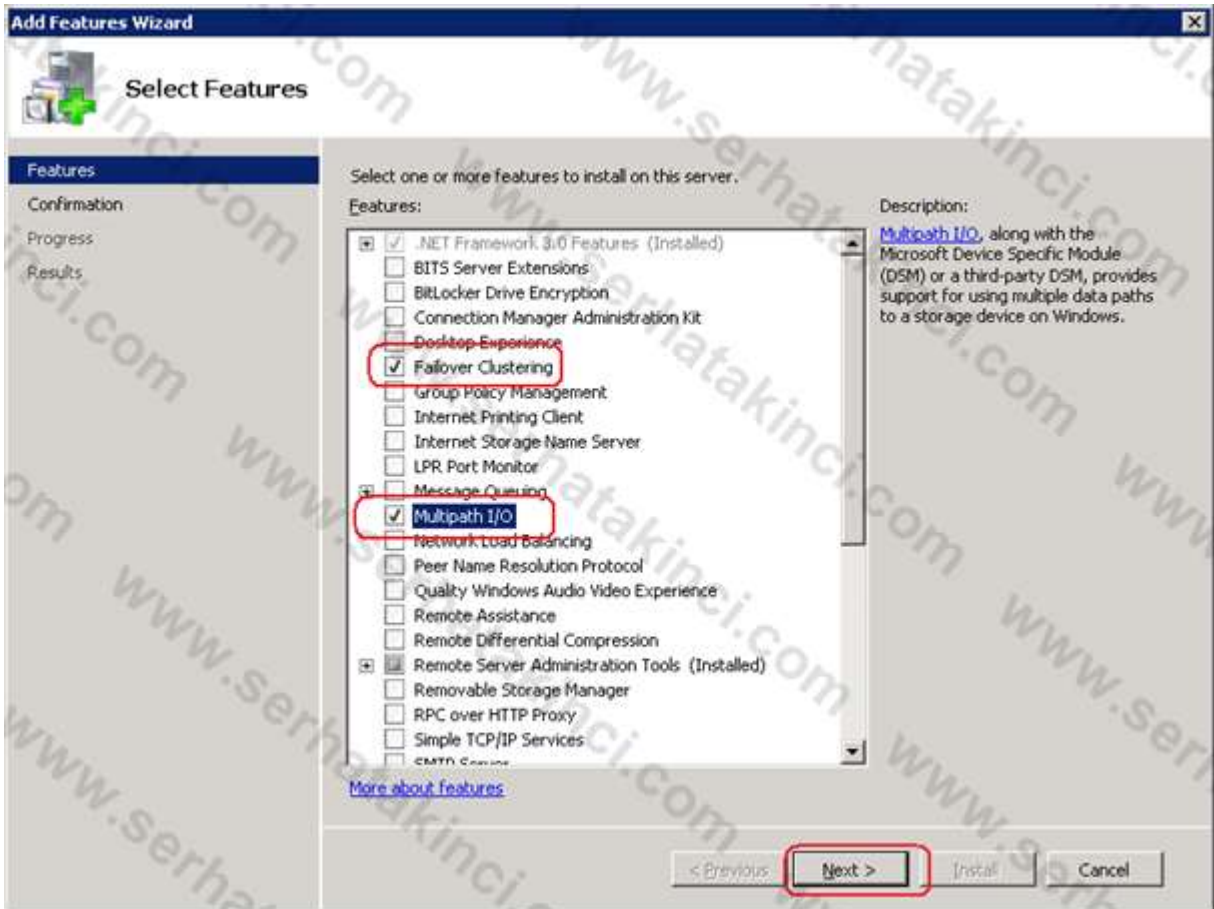
HV-Node1 için features yükleme işlemi tamam. Şimdi aynı işlemi diğer node yani HV-Node2 için yapmalıyız.

### 6.3. HV-Node2 Üzerine Gerekli Feature'ların Yüklenmesi

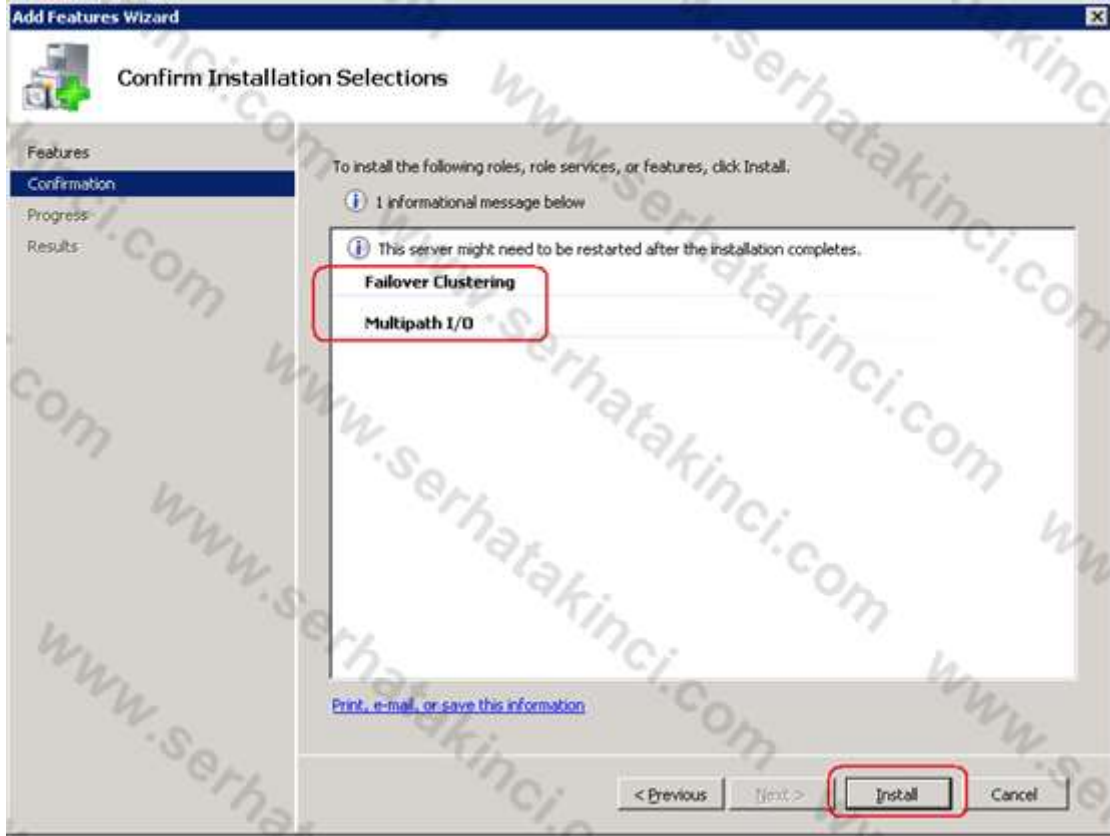
**HV-Node2** üzerinde **Server Manager** konsolunu açıyoruz ve **Features** bölümüne geçip **Add Features** diyoruz.



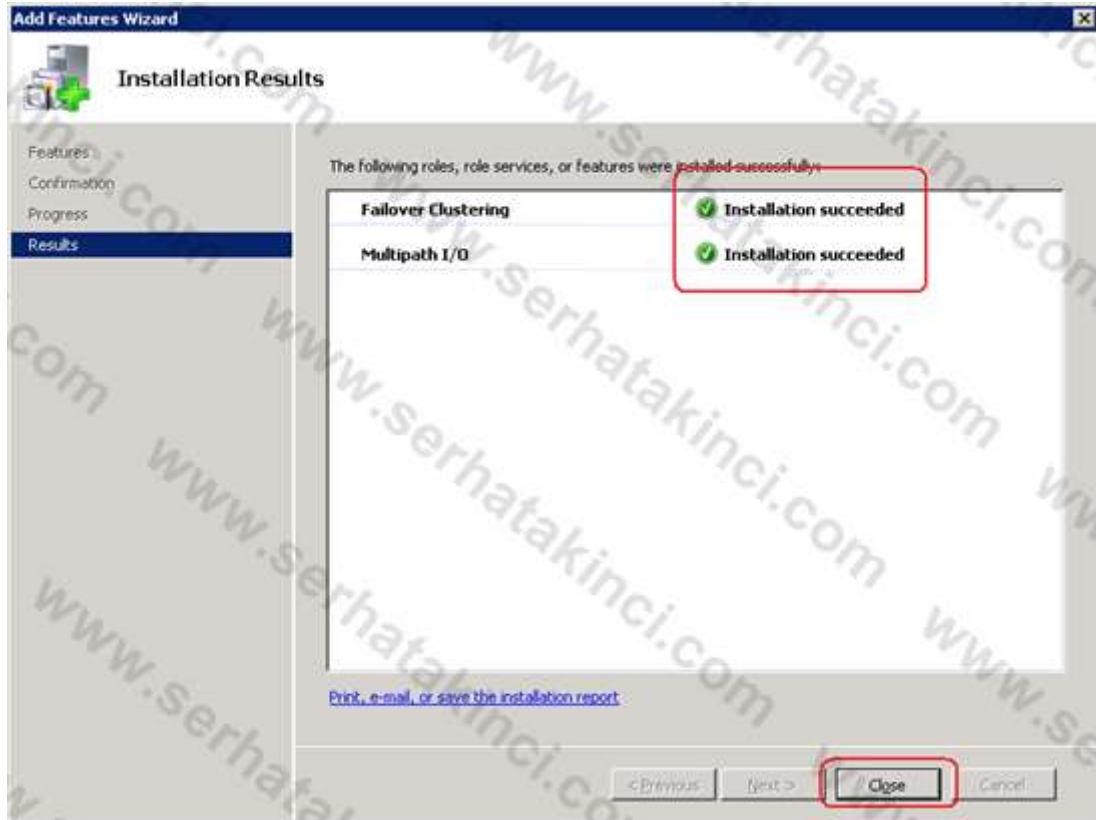
İlgili fetaure'ları seçiyoruz ve **Next** diyoruz.



**Install** diyerek yükleme işlemini başlatıyoruz.



**Succeeded** bilgisini gördükten sonra yükleme işlemi başarılı bir şekilde tamamlanmış oluyor.



**Close** ile pencereyi kapatıyoruz.

Böylece her iki node için temel konfigürasyonu yapmış, cluster için gerekli feature'ları yüklemiş olduk.

#### 6.4. iSCSI Initiator ile HV-Node1 için iSCSI Storage Discover

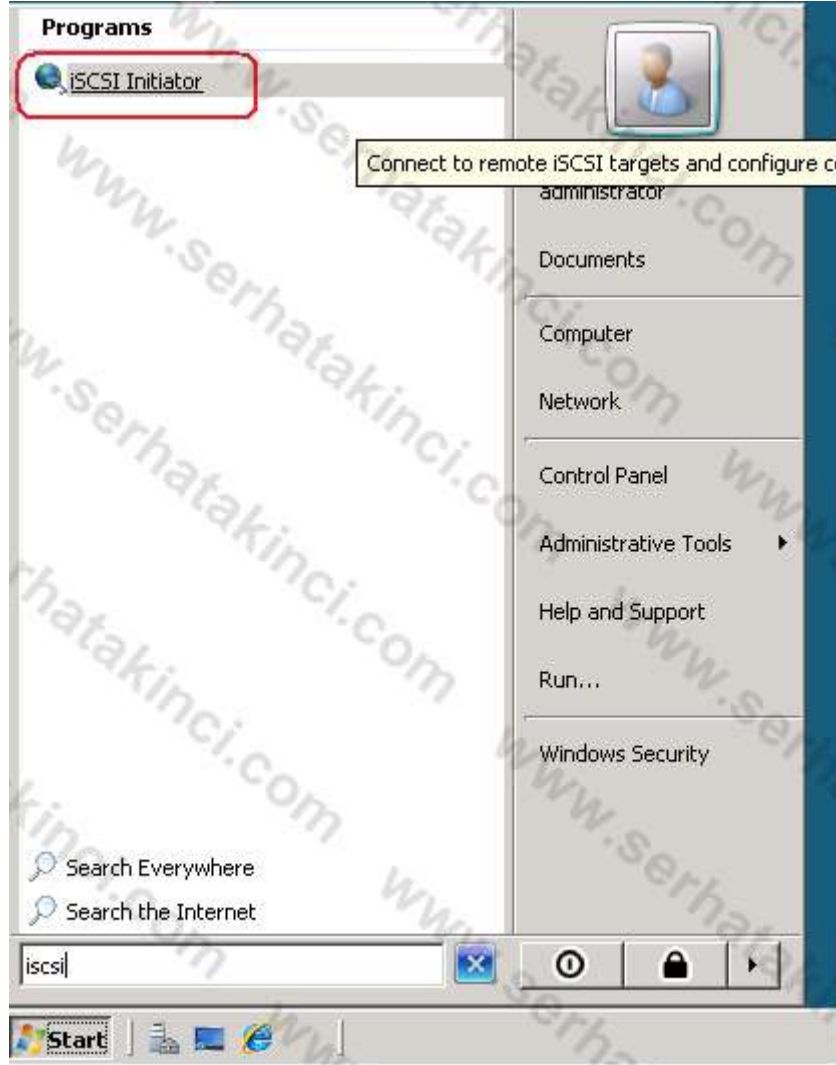
**Storage** makinesi üzerinde, **Microsoft iSCSI Software Target** ile iki **Virtual Disk** oluşturmuştuk.

Node'ların iSCSI Storage'e erişebilmeleri ve üzerindeki virtual diskleri kullanabilmeleri için ise **iSCSI initiator** yazılımını kullanacağız. **iSCSI initiator** Windows Server 2008 içerisinde yerleşik olarak gelen bir araçtır.

Virtual diskleri Nodelar üzerine atamadan önce storage'i göstermeli yani discover etmeliyiz. Daha sonra Virtual diskleri Nodelar üzerine atayabiliriz.

**HV-Node1** için **Discover** işlemiyle başlayalım.

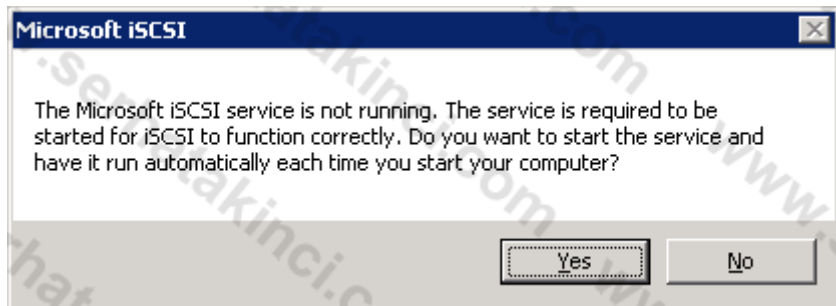
Öncelikle **iSCSI initiator**'ı buluyoruz.



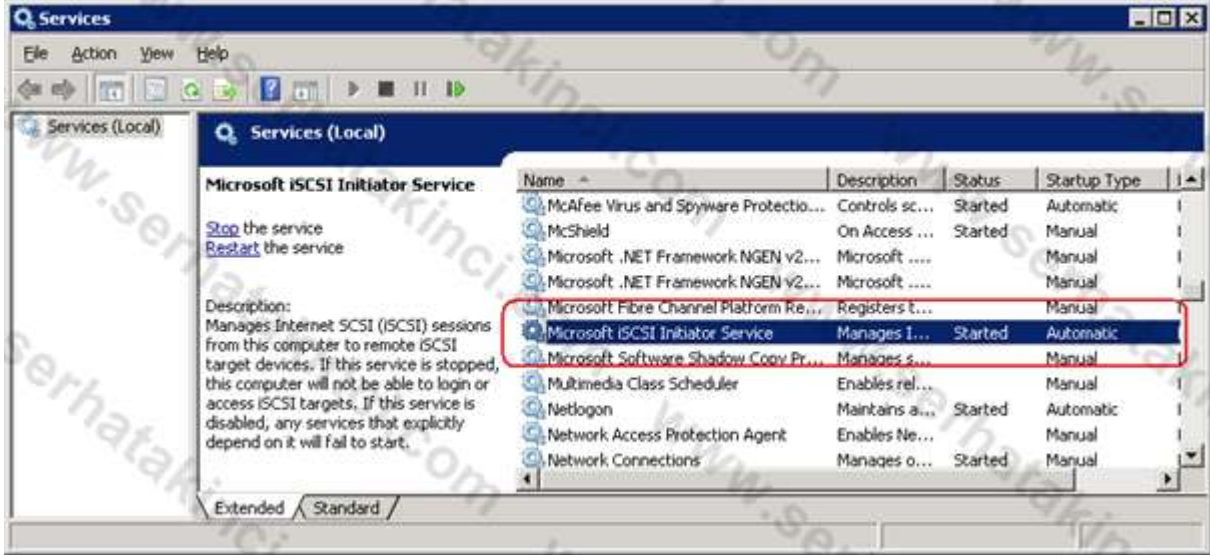
Çalıştırılım.

Gelen uyarıya **Yes** diyoruz çünkü **iSCSI initiator**'ı ilk kez çalıştırıyoruz ve **Microsoft iSCSI** servisi şu an stop durumda. Cluster yapısının sağlıklı çalışabilmesi için Microsoft iSCSI servisinin Node'lar restart olduğunda otomatik olarak start olması gerekiyor. Gelen uyarıya **Yes** diyerek bunu sağlamış oluyoruz.

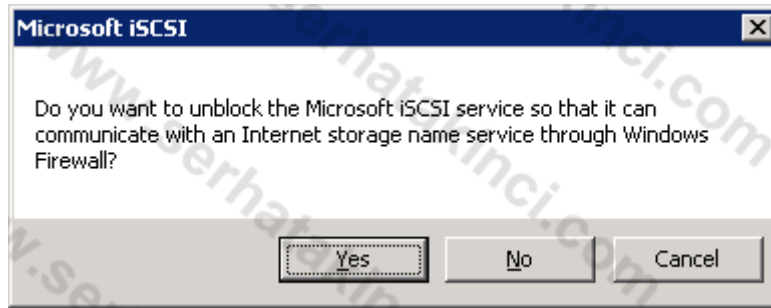
**Yes** dedikten sonra **Microsoft iSCSI Initiator Service** start oluyor ve sonraki sistem açılışlarında otomatik olarak başlayacak şekilde yapılandırılıyor.



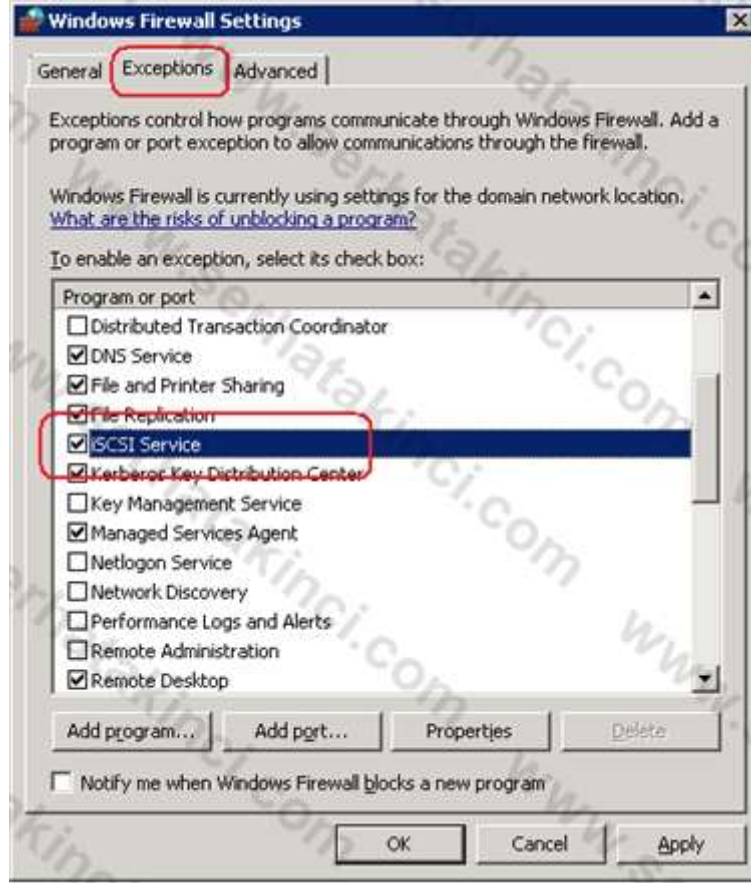




Eğer Hv-Node1 üzerinde **Windows Firewall** açıksa aşağıdaki gibi bir uyarı daha alırsınız. Servisin bloklanmaması için bu uyarıya da **Yes** diyoruz. Böylece **Windows Firewall** açık durumdayken iSCSI iletişimi sağlıklı bir şekilde çalışabilecek.

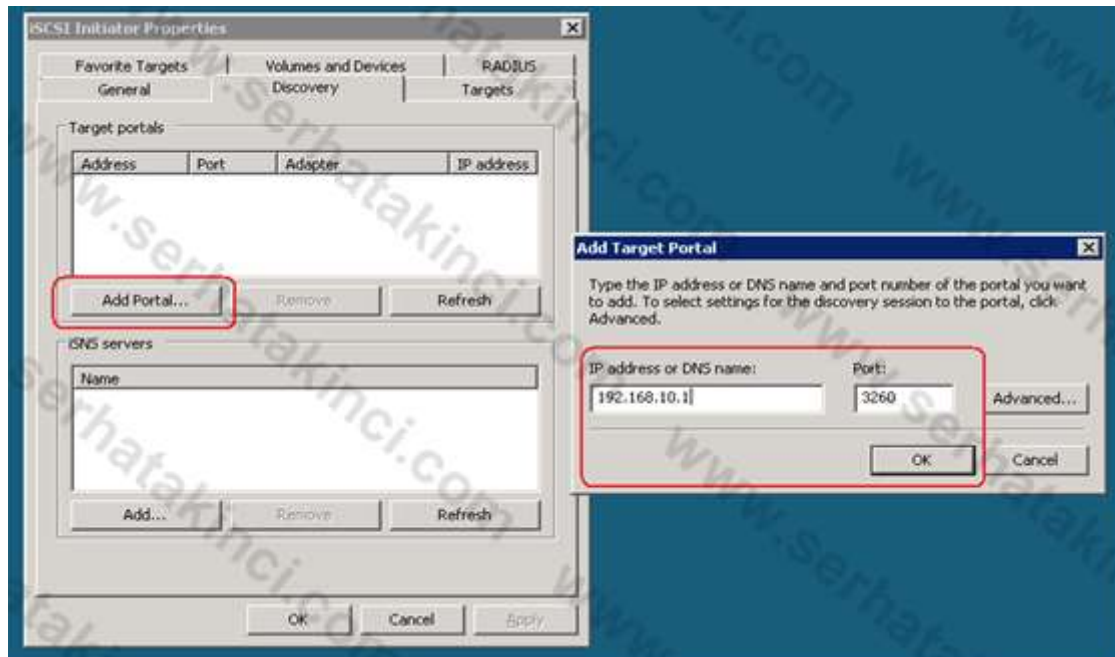


**Windows Firewall** üzerindeki bu tanımlı **Exceptions** tabından düzenleme şansınız da var.



iSCSI initiator açıldıktan sonra **Discovery** tabına geliyoruz ve **Add Portal...** diyerek storage'i ekliyoruz.

Bu işlem için IP adresi kullanabileceğiniz gibi storage'i temsil eden DNS Name de kullanabilirsiniz. Ben IP kullanıyorum ve burada yazdığım ip, storage'in iSCSI network üzerinde konuşan ip adresi (topology den hatırlayabilirsiniz – 192.168.10.1)

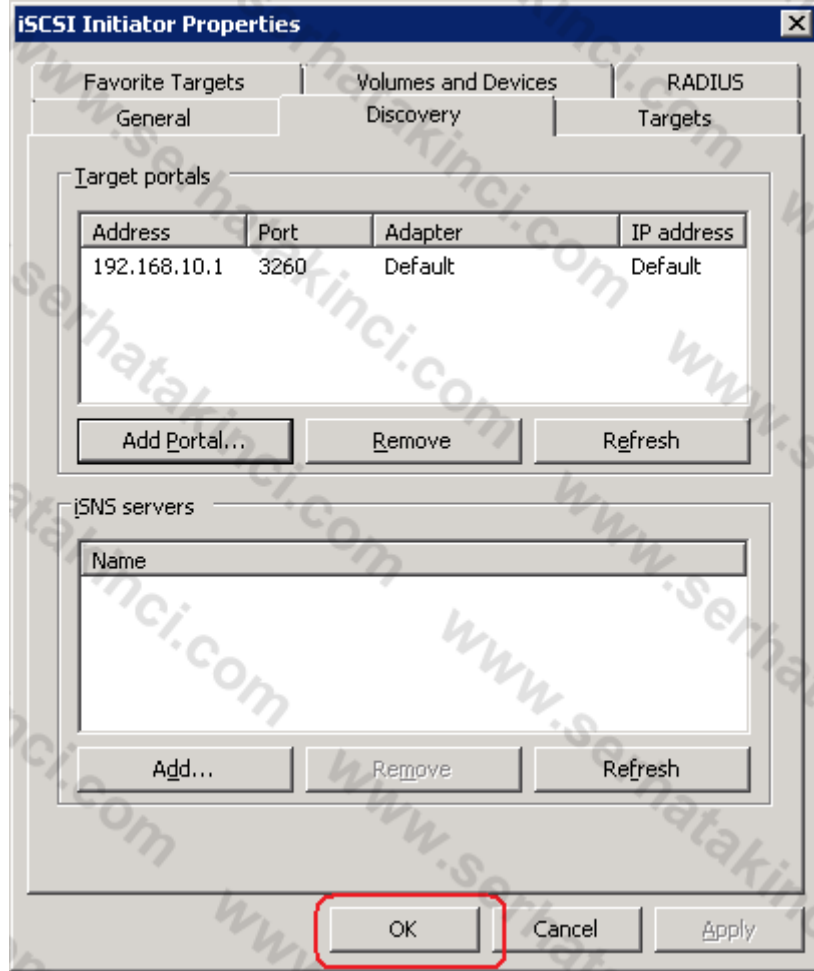




**Ok** diyerek pencereyi kapatıyoruz.

Eğer yanlış bir IP adresi yazarsanız ekleme işlemi gerçekleşmeyecektir.

Başarılı bir ekleme işleminden sonra aşağıdaki gibi görünüyor olmalı.

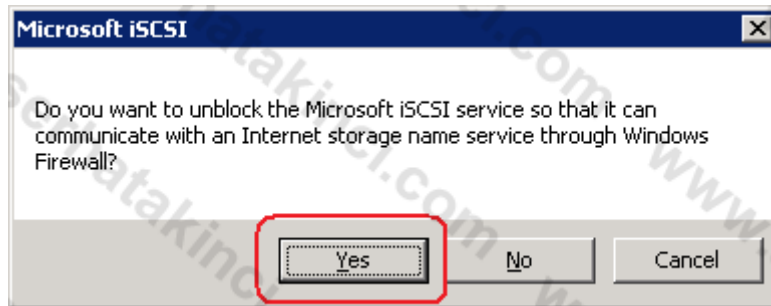
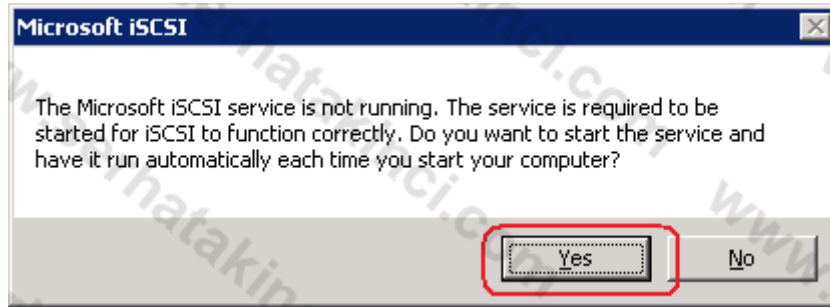


HV-Node1 üzerindeki Discover işlemi bu kadar. Az sonra storage üzerinde devam edeceğiz ancak öncelikle aynı işlemleri diğer node yani **HV-Node2** üzerinde de yapmamız gerekiyor.

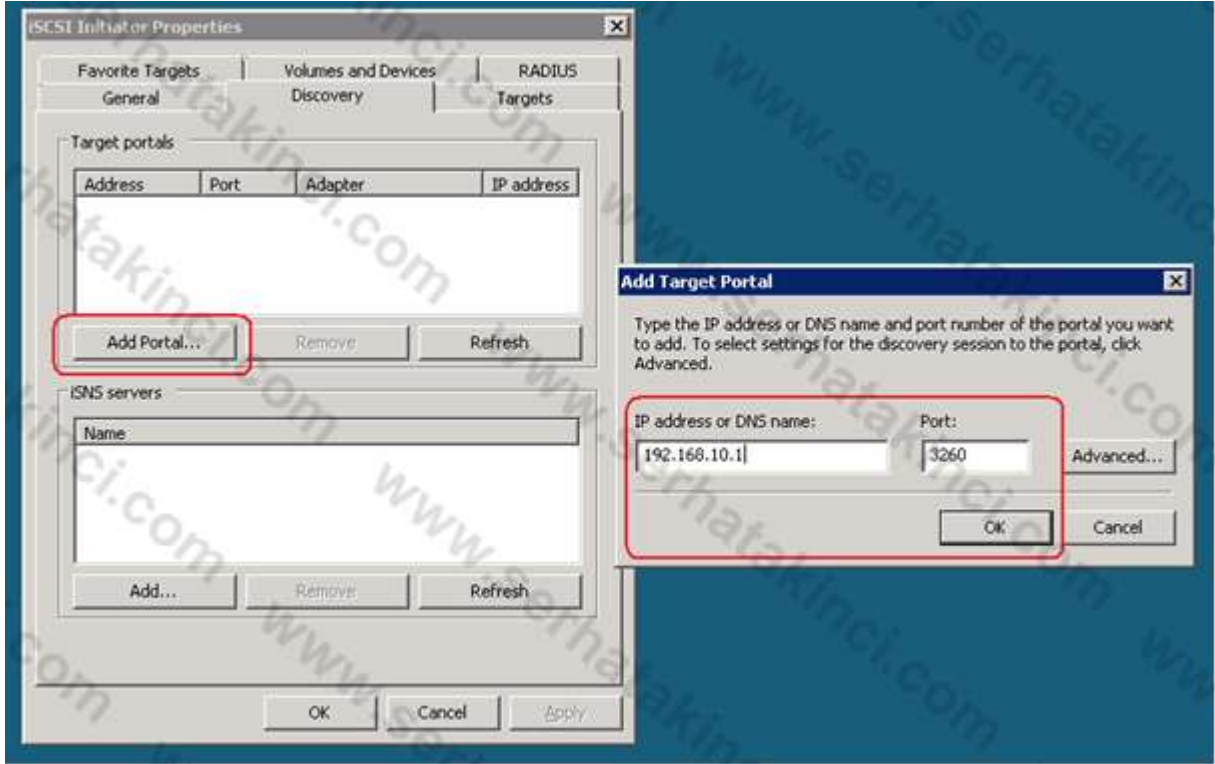
## 6.5. iSCSI Initiator ile HV-Node2 için iSCSI Storage Discovery

HV-Node1 için yaptığımız Discover işleminin aynısını HV-Node2 için de yapıyoruz.

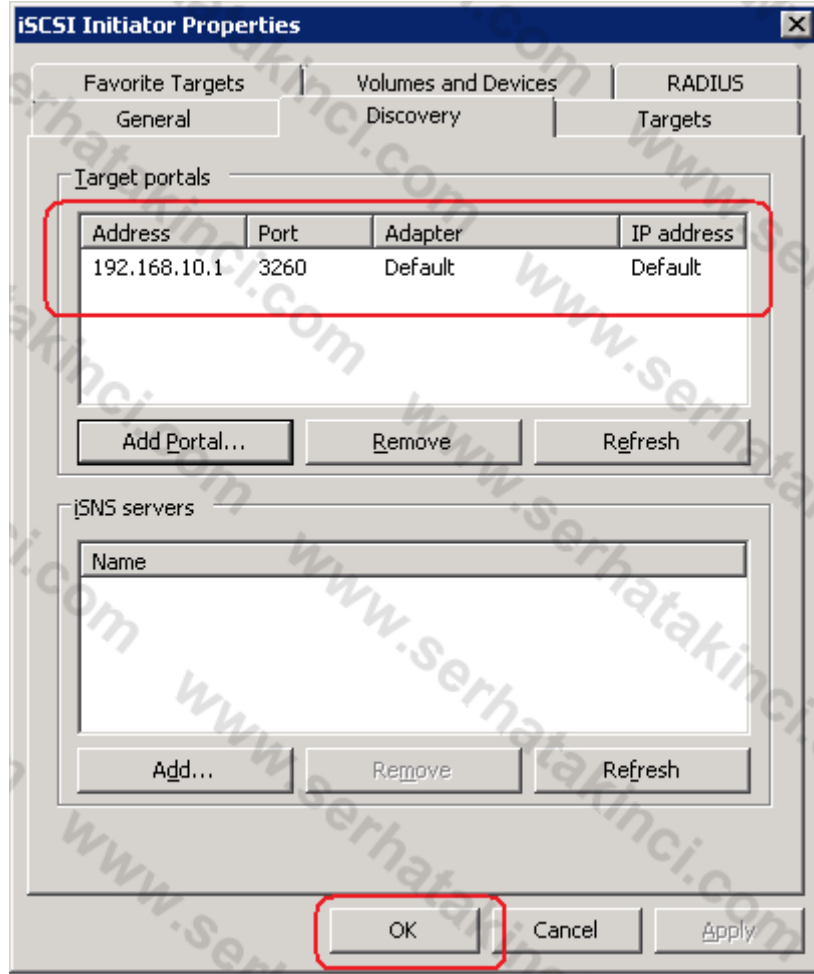
**HV-Node2** üzerinde **iSCSI initiator**'ı açıyoruz ve gelen uyarılara **Yes** diyoruz.



iSCSI initiator üzerinde **Discovery** tabında storage'i ekliyoruz.



Ve **Ok** diyerek bitiriyoruz.



Böylece her iki node'a storage'i göstermiş olduk. Ancak henüz Virtual Disk'leri node'lar üzerine bağlamadık.

Bu işlemi şu şekilde düşünebilirsiniz. Elimizde bir iSCSI Storage var ve biz Nodelara dedik ki "Bakın burada 192.168.10.1 ip adresli bir storage var ve eğer üzerinden sana disk atanırsa bunu kullanabilirsin"

Şimdi ise Node'ları storage üzerinde **Target** olarak tanımlayacağız ve önceden yarattığımız Virtual Disk'leri bu node'lara bağlayacağız.

## 7. Cluster Nodeları için Disk Atamak

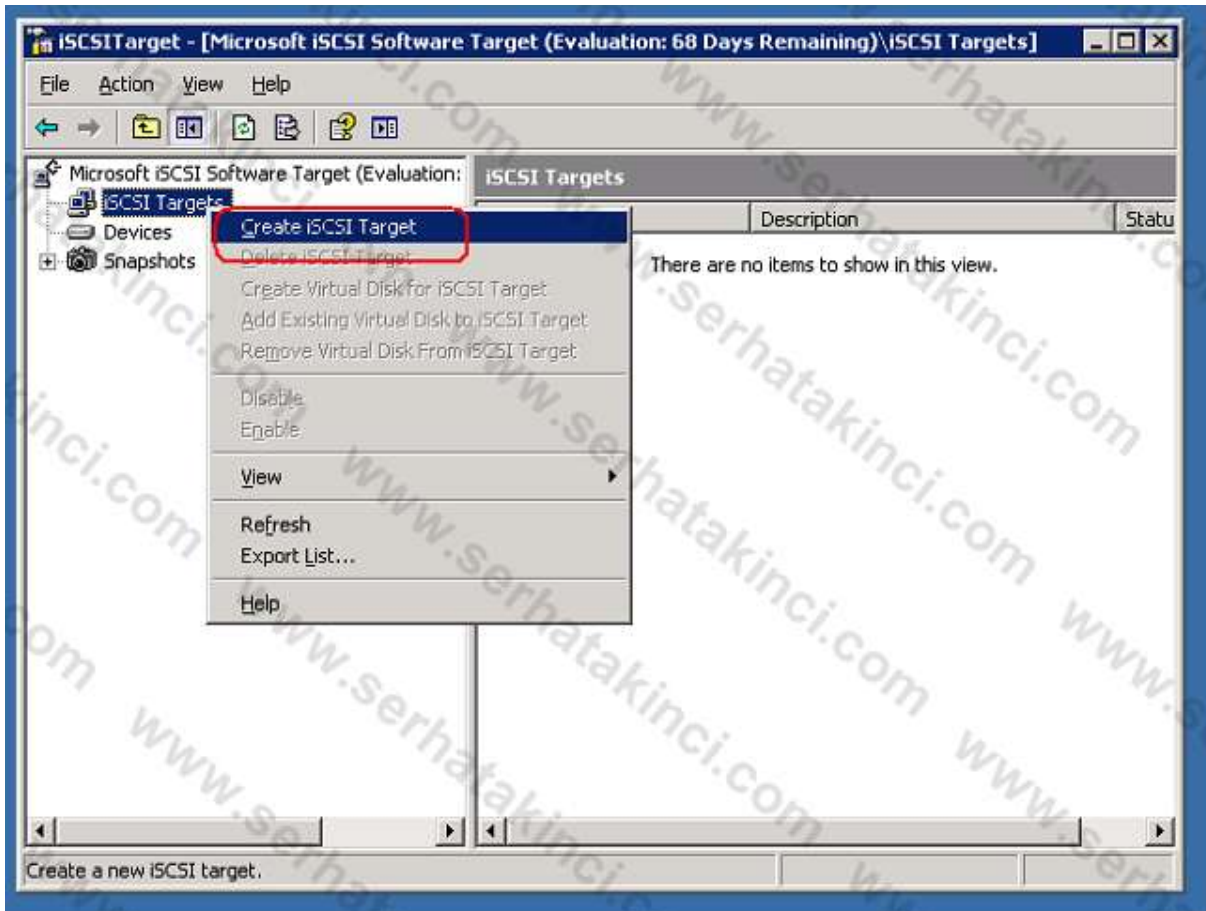
Storage yani organizasyonumuzda **storage.bemar.corp** olarak yer alan ve **Microsoft iSCSI Software Target** yazılımını çalıştıran makineye geçiyoruz.

**Microsoft iSCSI Software Target** yönetim konsolunu açalım.

İlk işlemimiz az önce storage'i gösterdiğimiz Node'lar için **Target** tanımlamak. Bunu Nodelara storage'i gösterme işleminin tam tersi olarak düşünebilirsiniz. Yani bu sefer de Storage'e diyoruz ki "Bak burada sana gelen iki adet node var ve bunları senin üzerinde target olarak tanımlayalım ki atayacağımız diskleri kullanabilsinler :)"

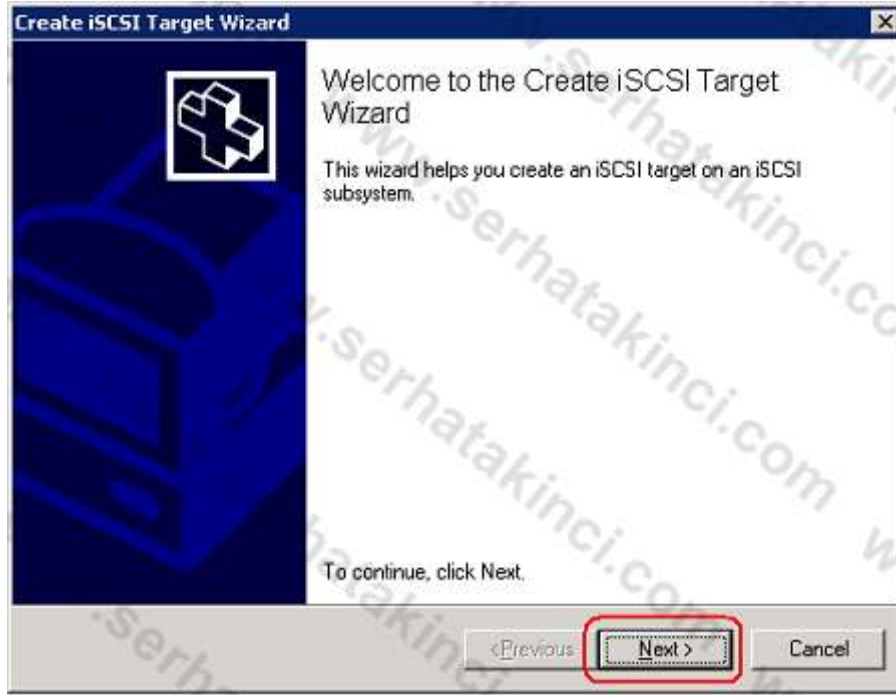
### 7.1. iSCSI Storage üzerinde HV-Node1 için Target Tanımı

Storage makinesinde iSCSITarget konsolu üzerinde **Create iSCSI Target** diyerek **Create iSCSI Target Wizard**'i açıyoruz.



**Next** ile devam ediyoruz.





İlk target için bir isim giriyoruz: **HV-NODE1**

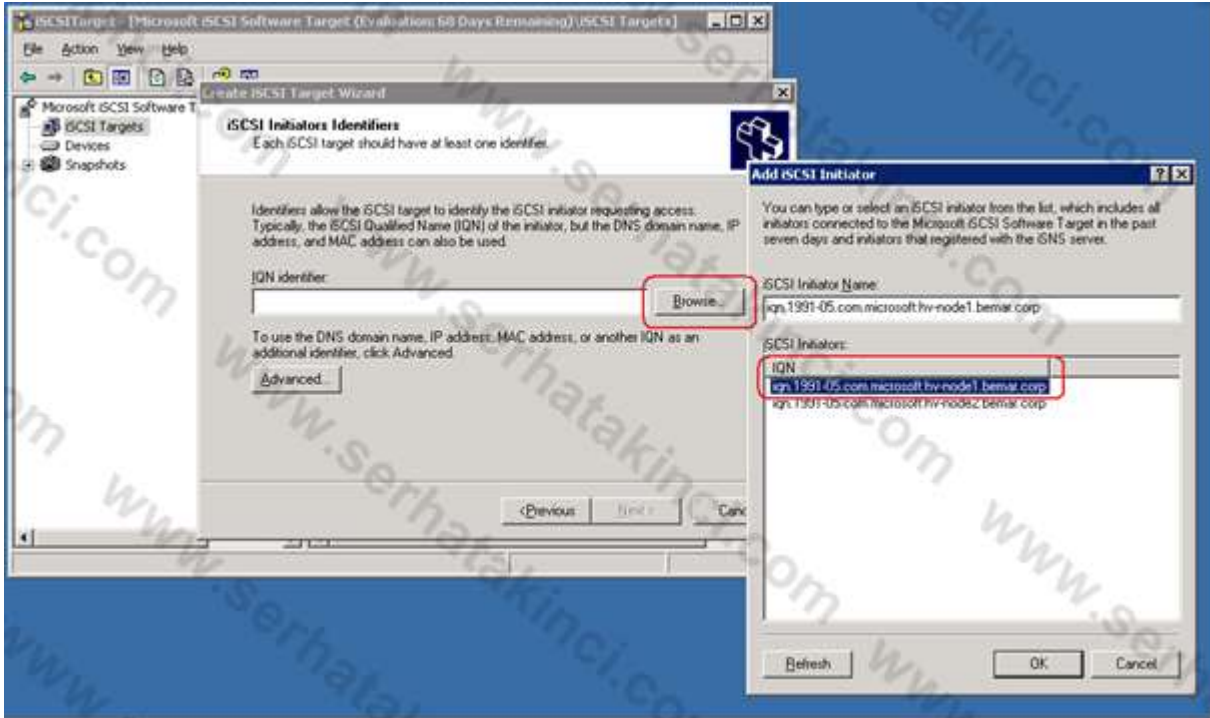


Sonraki pencerede ilgili target için **IQN identifier** tanımlıyoruz.

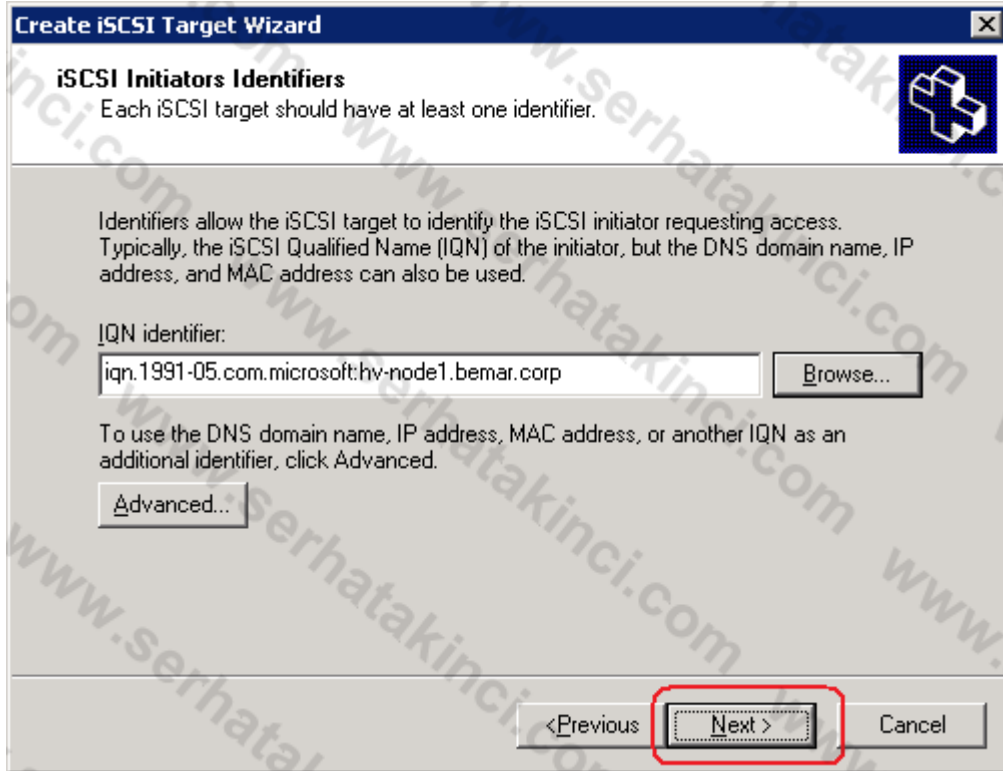
**Browse** butonuna tıkladığımızda iki farklı **IQN** listeleniyor olacak. Bu IQN'ler, az önce iSCSI initiator ile storage'e bağladığımız (storage'i gösterdiğimiz) node'ların kendisidir. Yani node'lar üzerinde yaptığımız **discover** işleminin amacı buydu. Eğer o işlemi yapmasaydık, IQN listesinde bu iki node'u göremeyecektik.

İlk target ekleme işlemini HV-Node1 için yapıyoruz, bu nedenle **Hv-node1.bemar.corp** olan'ı yani ilk node'u temsil eden IQN'i seçerek devam ediyorum.

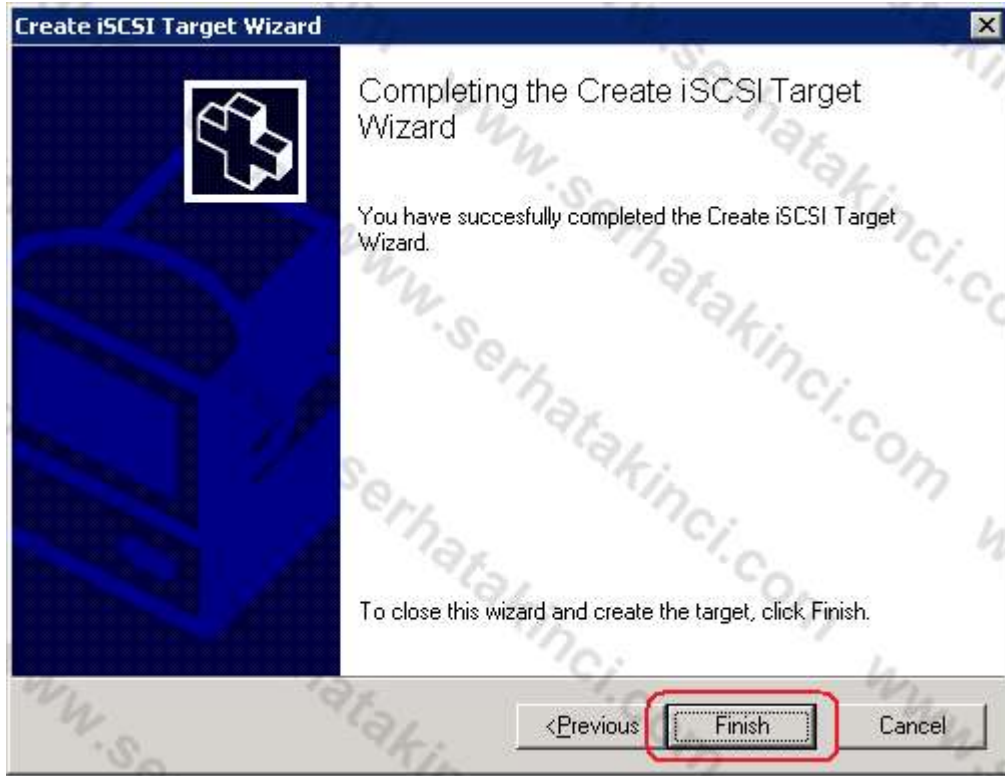
iqn.1991-05.com.microsoft.hv-node1.bemar.corp



**OK** dedikten sonra **Next** ile devam ediyoruz.



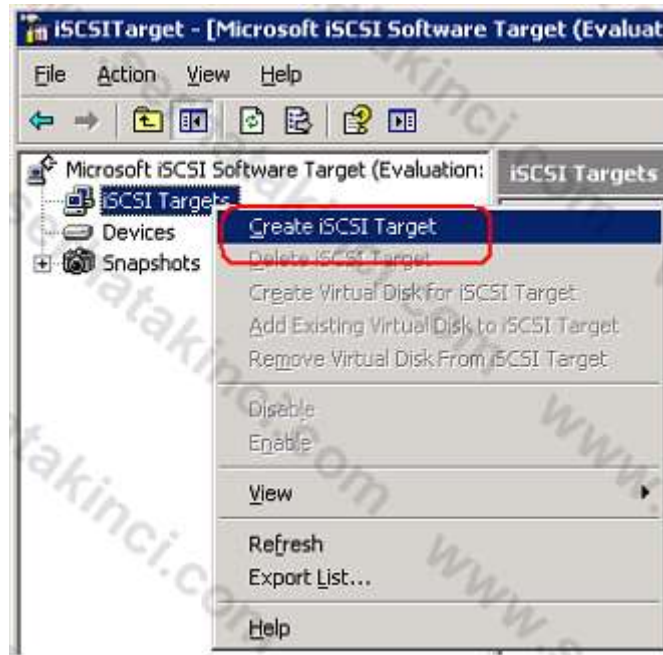
Ve **Finish** ile Target ekleme işlemini bitiriyoruz.



HV-Node1 için target ekleme işlemi bitti. Şimdi aynı işlemi **HV-Node2** için de yapalım.

## 7.2. iSCSI Storage üzerinde HV-Node2 için Target Tanımı

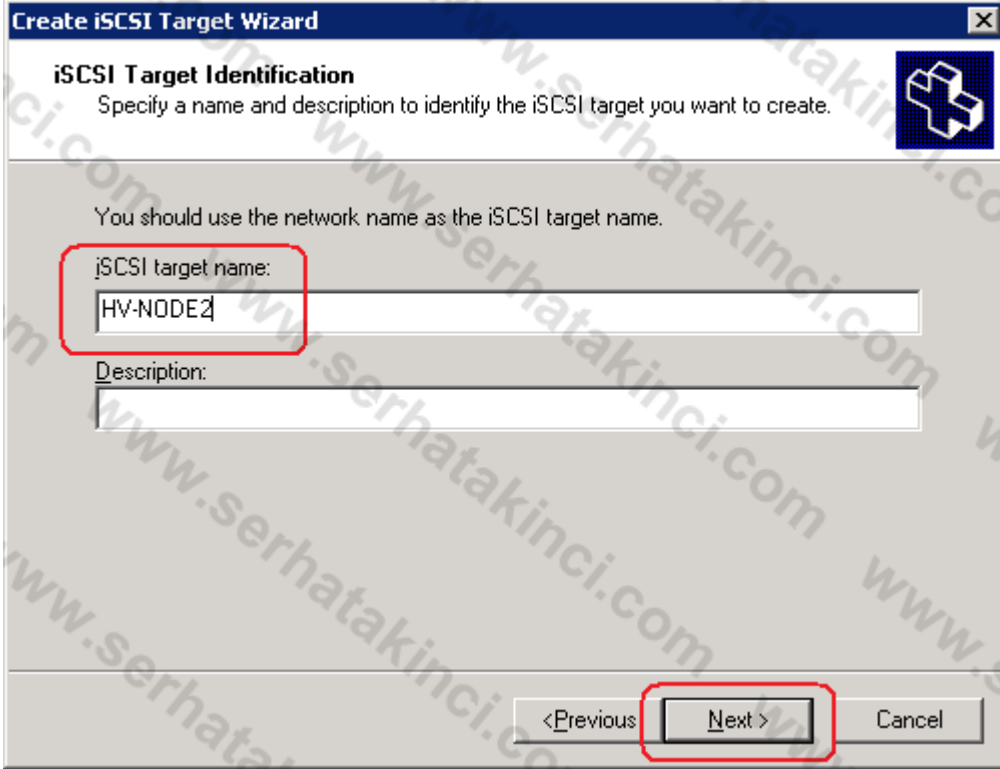
Aynı şekilde **iSCSI Targets**'a sağ tıklayıyoruz ve **Create iSCSI Target** diyerek wizard'ı açıyoruz.



**Next** ile ilerliyoruz.



İkinci target için bir isim giriyoruz: **HV-NODE2**



The image shows a screenshot of the 'Create iSCSI Target Wizard' window, specifically the 'iSCSI Target Identification' step. The window has a blue title bar with the text 'Create iSCSI Target Wizard' and a close button. Below the title bar, the text 'iSCSI Target Identification' is displayed, followed by the instruction 'Specify a name and description to identify the iSCSI target you want to create.' A small icon of a server with a plus sign is visible on the right. The main area of the window contains the text 'You should use the network name as the iSCSI target name.' Below this, there are two text input fields. The first field is labeled 'iSCSI target name:' and contains the text 'HV-NODE2'. The second field is labeled 'Description:' and is empty. At the bottom of the window, there are three buttons: '<Previous', 'Next >', and 'Cancel'. The 'Next >' button is highlighted with a red rectangle.

Sonraki pencerede ilgili target için **IQN identifier** seçiyoruz.

Yine **Browse** butonuna tıkladığımızda iki farklı **IQN** listeleniyor olacak. Biri az önce tanımladığımız HV-Node1'e ait. Diğeri ise HV-Node2'ye ait.

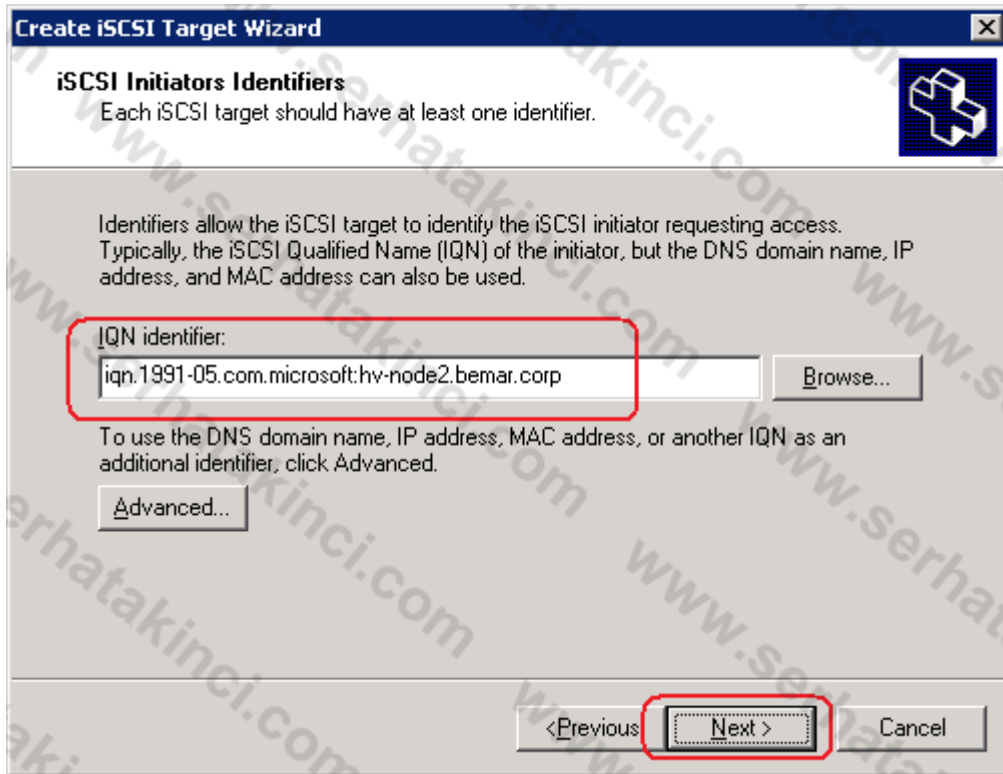
Bu seferki ekleme işlemini HV-Node2 adına yaptığımız için **Hv-node2.bemar.corp** olan'ı yani ikinci node'u seçerek devam ediyorum.

iqn.1991-05.com.microsoft.**hv-node1.bemar.corp**

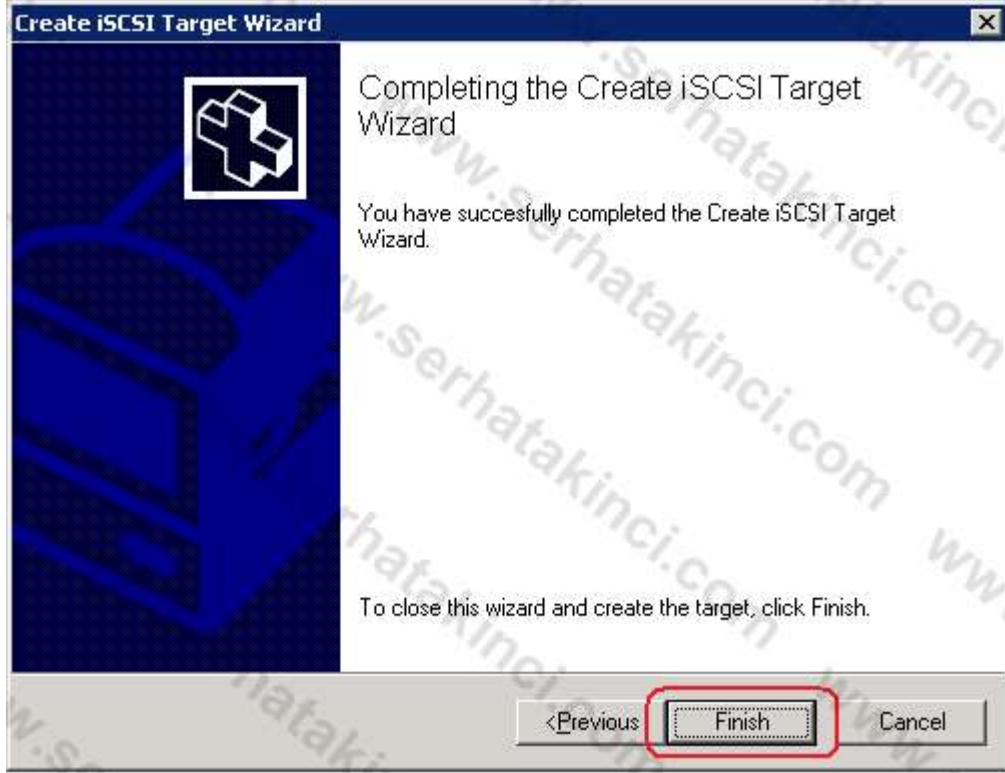




Next ile devam ediyoruz.

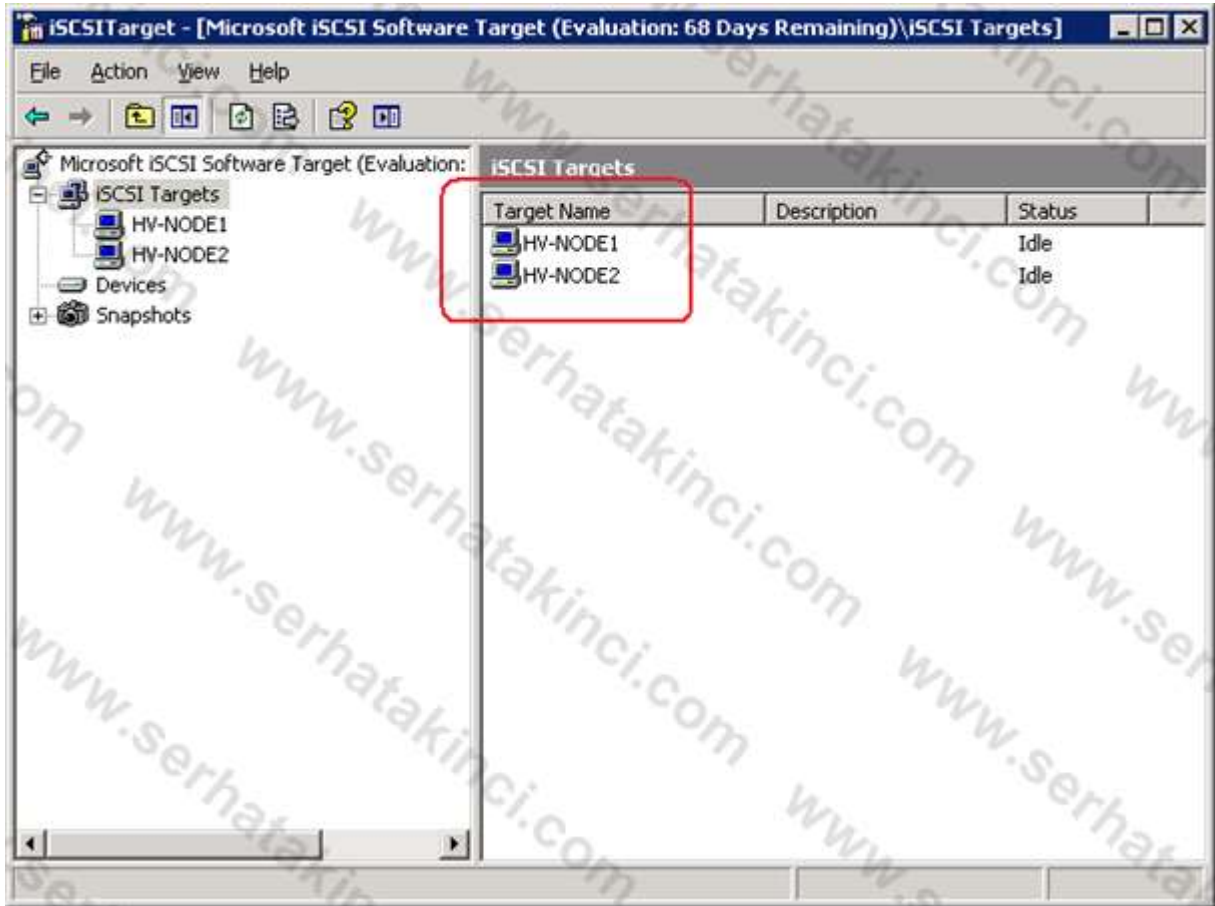


Ve **Finish** ile ikinci target ekleme işlemini de bitiriyoruz.



Target ekleme işlemi bu kadar.

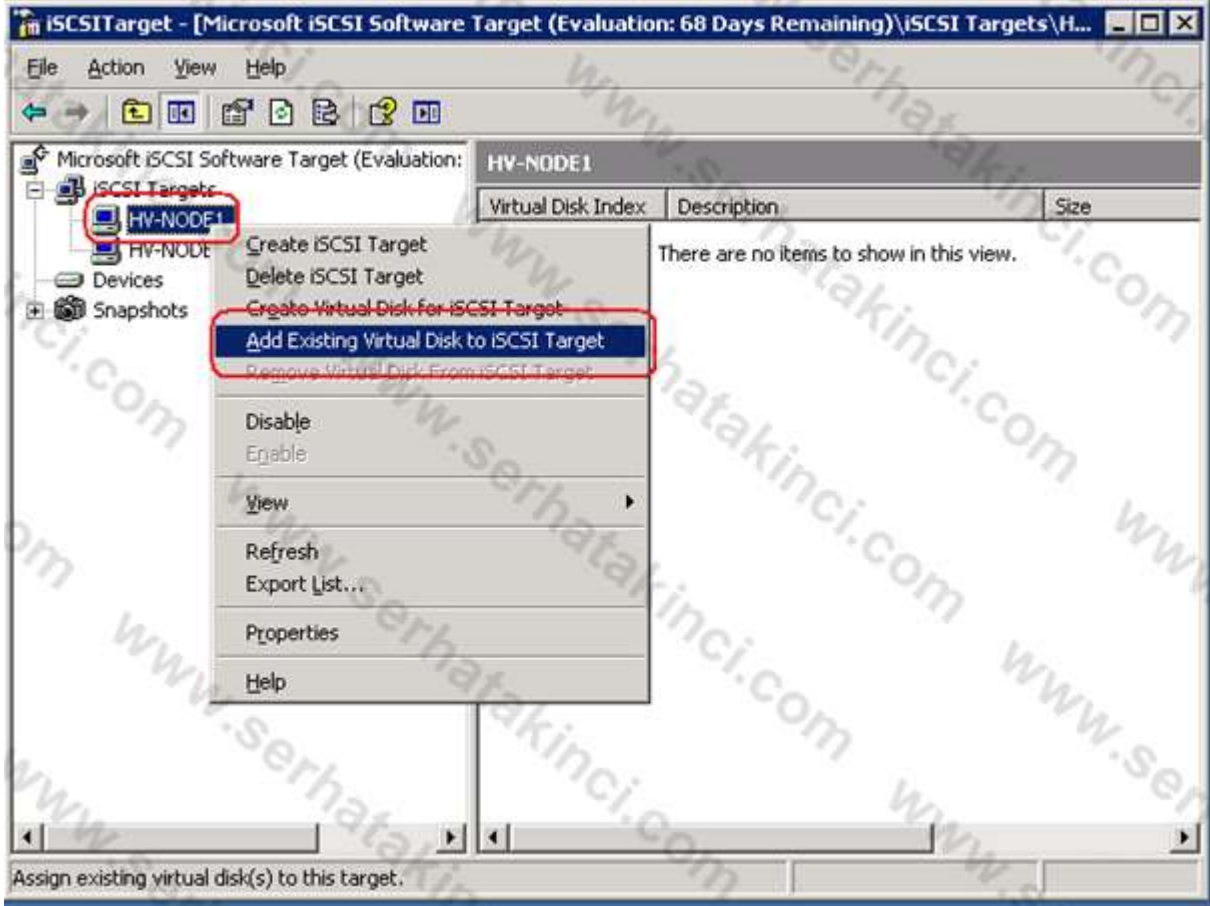
Her iki node'u iSCSI Target olarak tanımladığımıza göre, konsol üzerindeki görüntü aşağıdaki gibi olmalı.



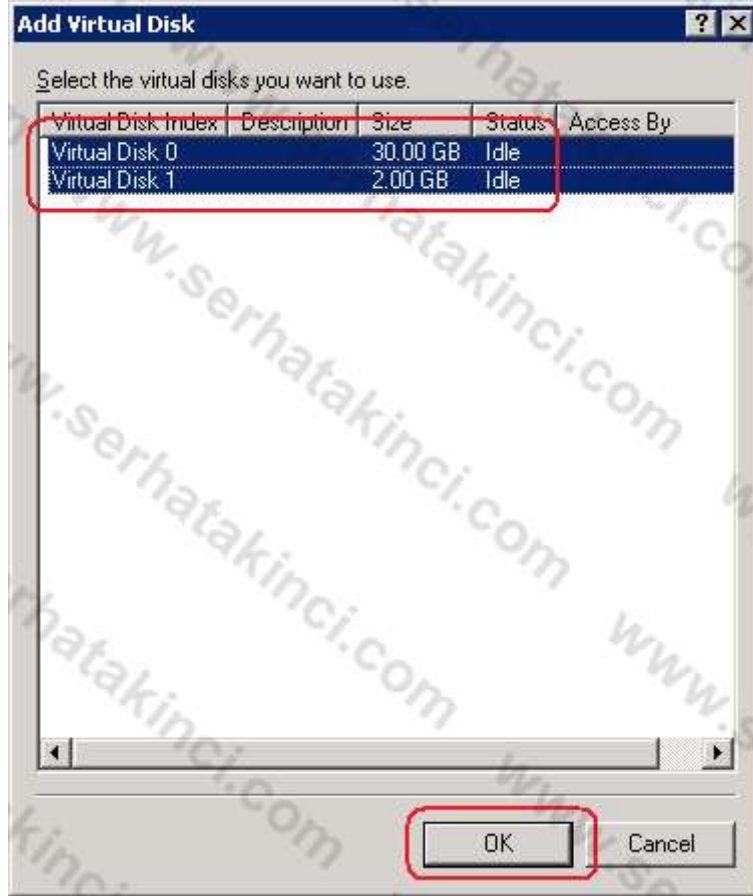
Artık eklediğimiz bu target'lara yani cluster node'larına Virtual Disk'ler ekleyebiliriz.

### 7.3. iSCSI Storage üzerinde HV-Node1 için Disk (virtual disk) Ekleme

iSCSI Targets bölümünde HV-Node1'e sağ tıklıyoruz ve **Add Existing Virtual Disk to iSCSI Target** diyoruz.



Ekleme istediğimiz **Virtual Disk**'leri seçeceğiz. Henüz yarattığımız iki adet disk olduğu için, listede iki Virtual Disk yer alıyor. Yazımızın ilk bölümlerinde uyguladığımız yöntemle yeni Virtual Disk'ler yaratabilir, bu şekilde target'lar'a ekleyebilirsiniz.

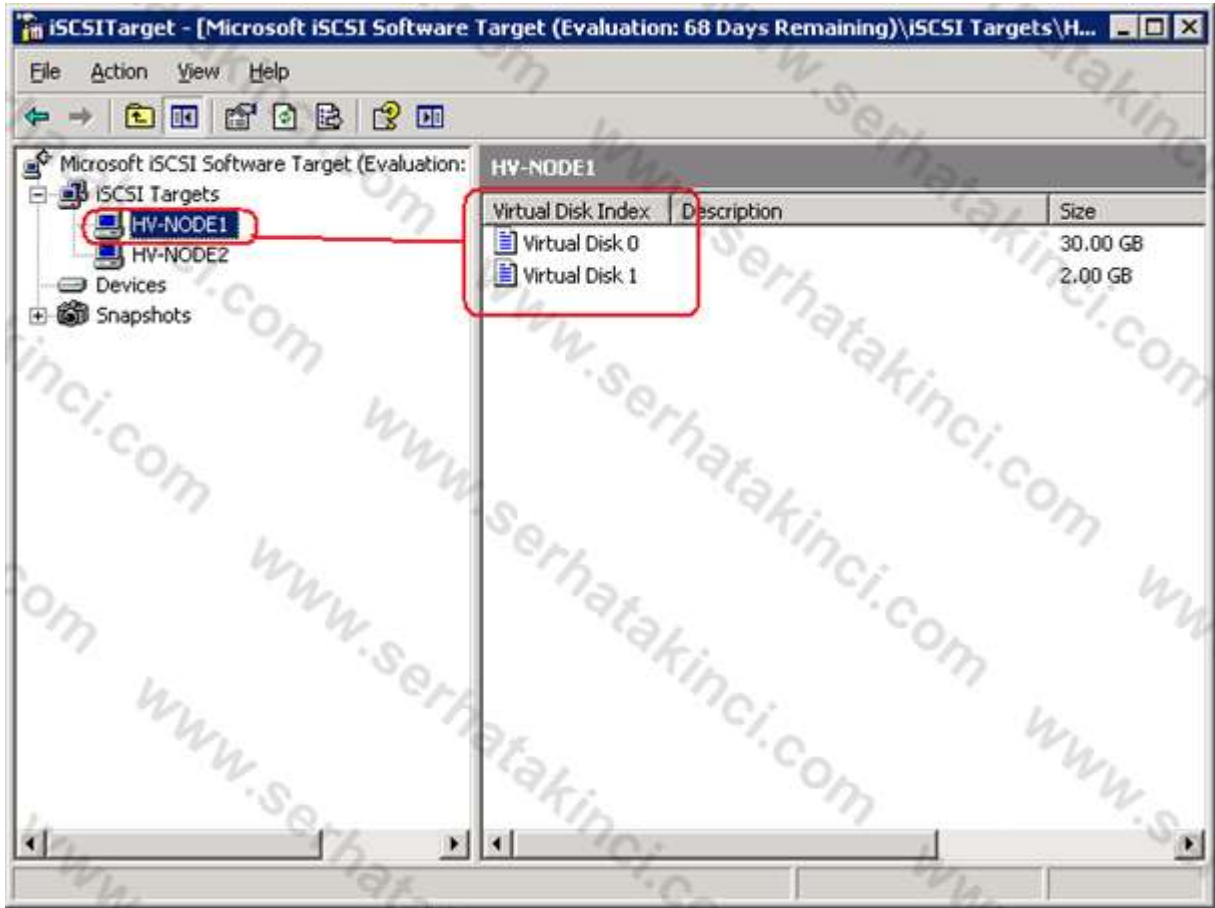


Bir diski **Quorum**, diğer diski ise **sanal makineler** için kullanacağız. Bu nedenle iki adet virtual disk yaratmıştık ve ikisini de seçtik.

**Ok** diyerek işlemi tamamlıyoruz.

Konsol üzerinde aşağıdaki gibi görünüyor olmalı.

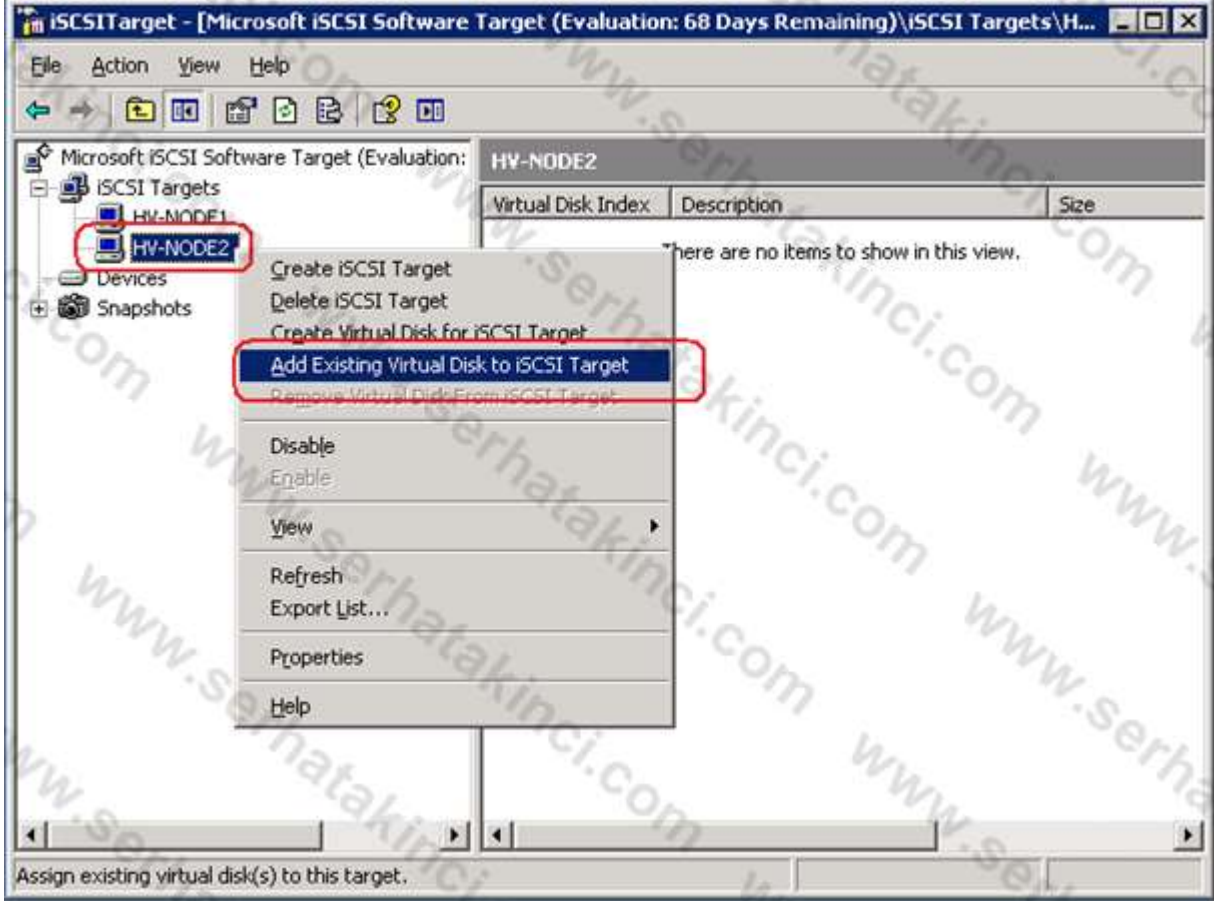




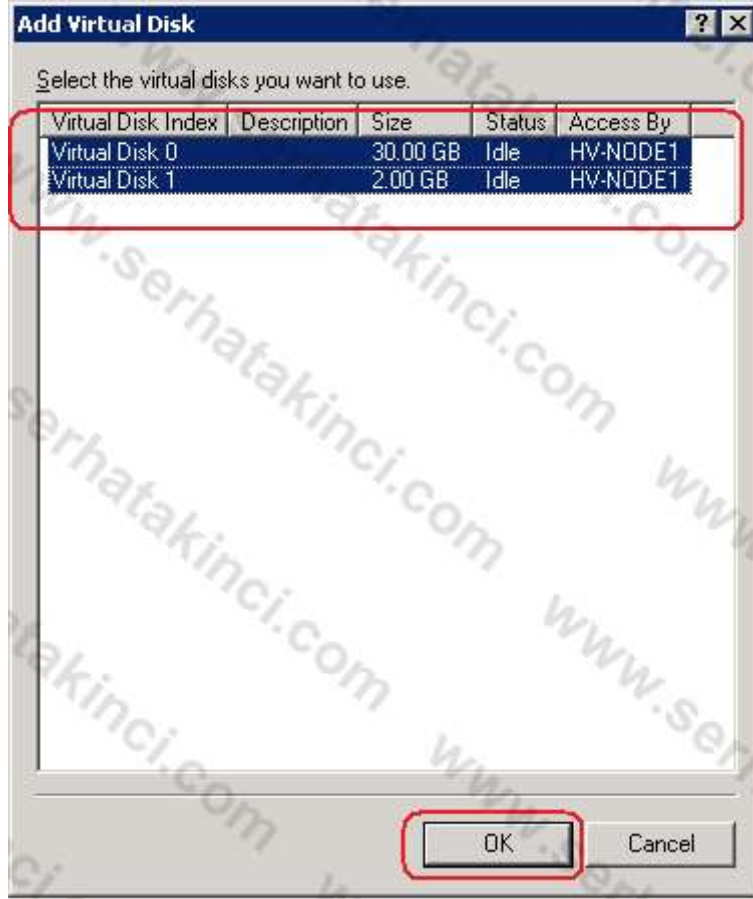
Aynı işlemi **HV-Node2** için de yapmamız gerekiyor.

#### 7.4. iSCSI Storage üzerinde HV-Node2 için Disk (virtual disk) Ekleme

iSCSI Targets bölümünde bu sefer HV-Node2'ye sağ tıklıyoruz ve **Add Existing Virtual Disk to iSCSI Target** diyoruz.

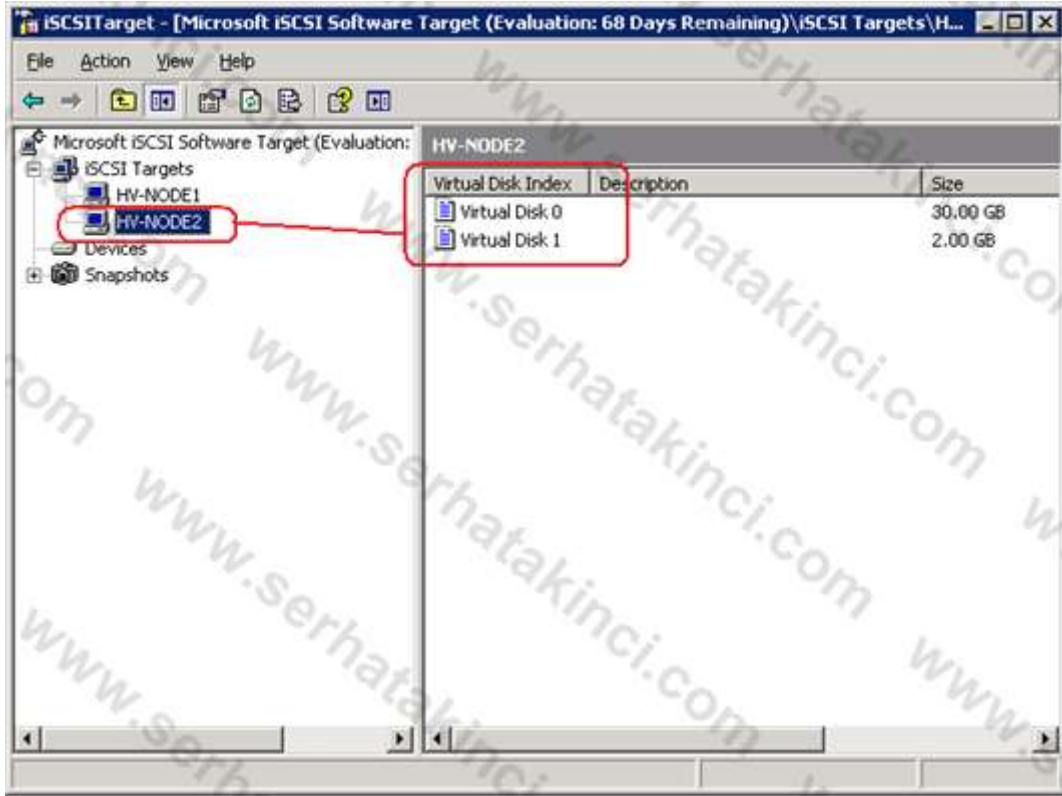


İki Virtual Disk'i de seçip **Ok** diyoruz.



Dikkat ederseniz yukarıdaki pencerede **Access By** bölümünde HV-Node1 var çünkü bu diskleri az önce HV-Node1'e eklemiştik. Bu işlemten sonra buraya HV-Node2 de gelecek.

Ekleme işleminden sonra konsolda aşağıdaki gibi görünüyormalı.



**Devices** kısmında ise her iki Virtual Disk'in, her iki node tarafından erişilebileceği bilgisini görüyor olmalıyız.



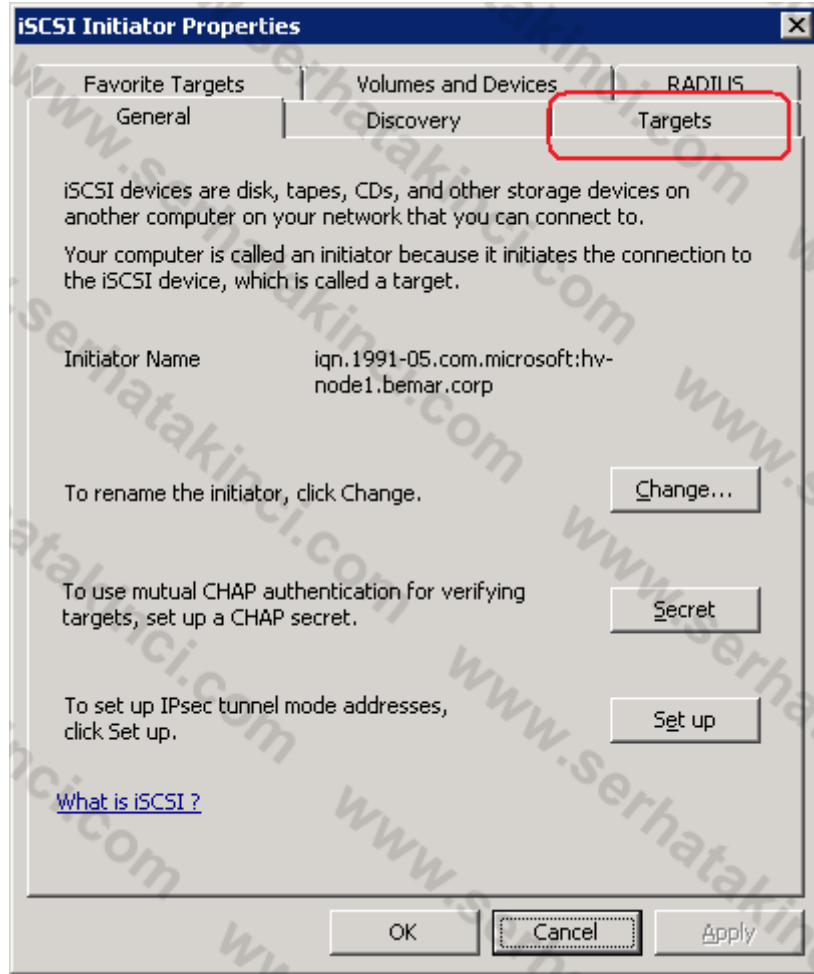
Her şey örtüşüyor ise **storage** üzerindeki işimiz bitti diyebiliriz :)

## 7.5. HV-Node1 üzerinde Disk (virtual disk)'i Görmek

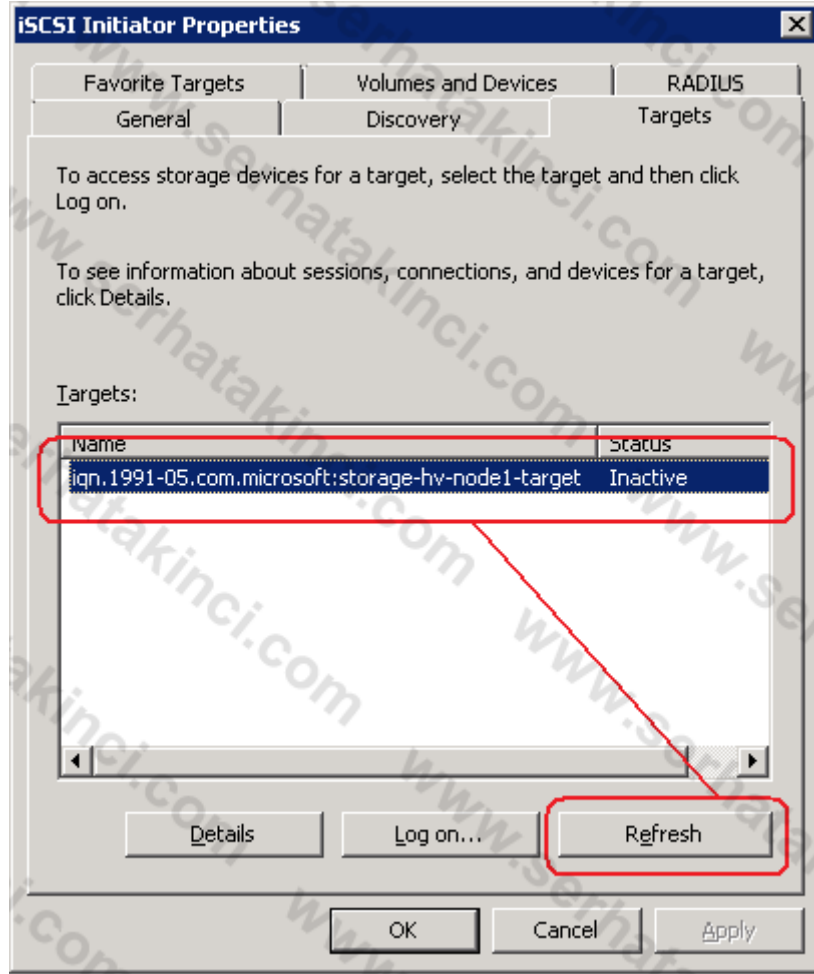
Node'ları storage üzerinde target olarak tanımlamıştık ve önceden yarattığımız Virtual Disk'leri bu target'lara eklemiştik. Bu Virtual Disk'leri fiziksel node'lar üzerinde kullanabilmek için, yine node'lar üzerinde çalışan iSCSI initiator yazılımında son bir konfigürasyon daha yapmamız ve Virtual Disk'leri fiziksel node'lara göstermemiz gerekiyor.

Bu işlemi yine her iki node üzerinde de yapacağız.

Öncelikle **HV-Node1**'e gidiyoruz ve **iSCSI initiator** yazılımını açıyoruz.



**Targets** tabına geçip **refresh** diyoruz ve bu node için oluşturduğumuz target'ın listede görünmesini sağlıyoruz.



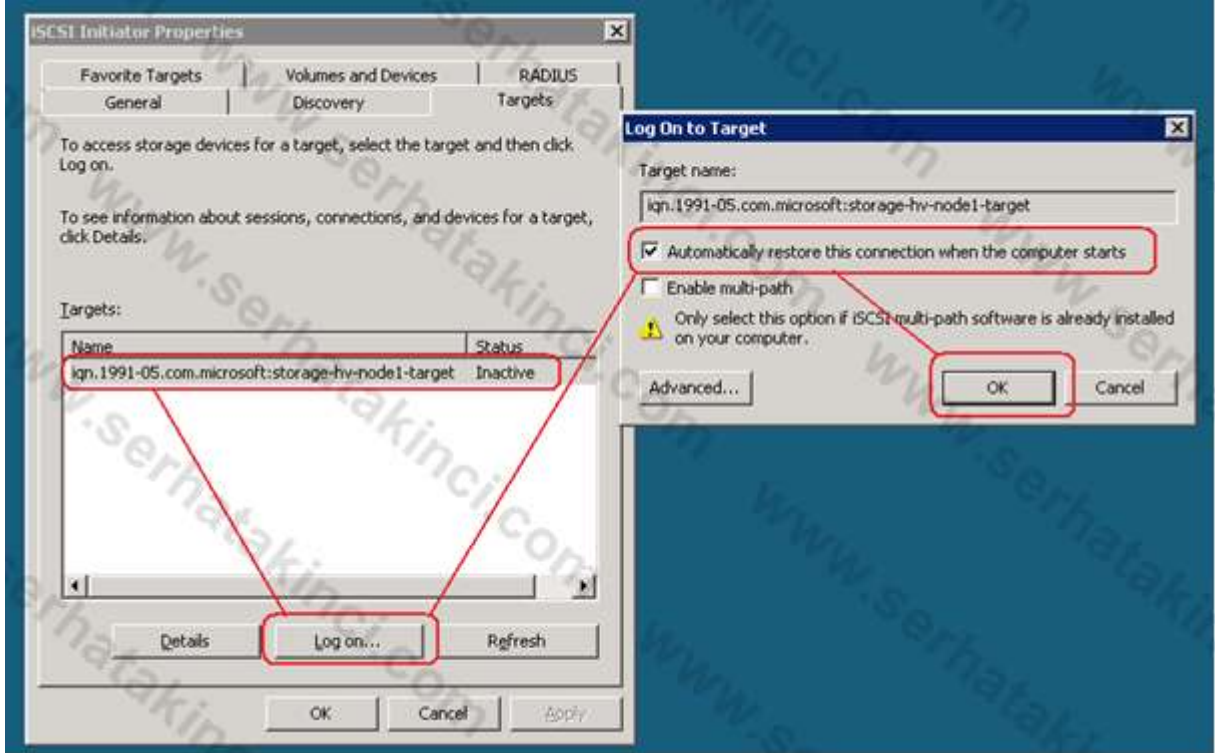
Eğer **Refresh** dediğinizde Target gelmiyor ise, storage üzerinde yaptığınız target tanımında yada storage'e erişimde bir sorun vardır.

Yukarıda listelenen target'ın durumu **inactive** olarak görünüyor. Bu durum şimdilik normaldir.

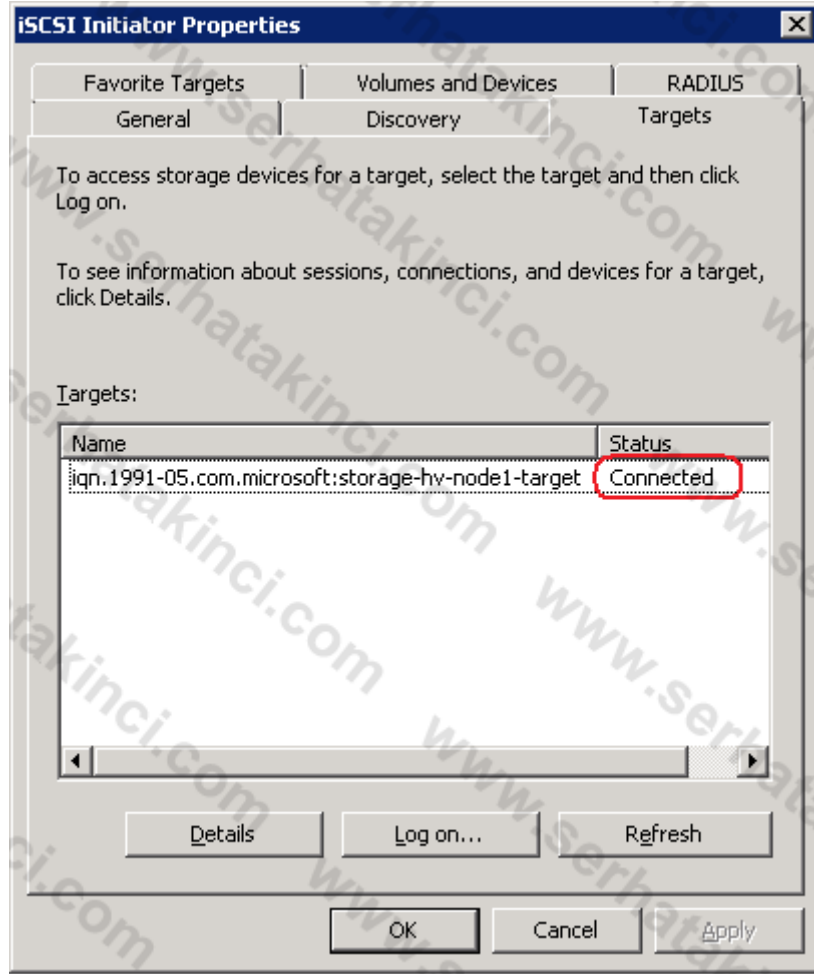
**Target**'ı seçtikten sonra **Log On..** diyoruz ve gelen pencerede **Automatically restore this connection when the computer starts** kutucuğunu tıklayıp **Ok** diyoruz.

Bu ayar ile sistem başladığında ilgili target bağlantısının otomatik olarak yapılmasını sağlamış olduk.



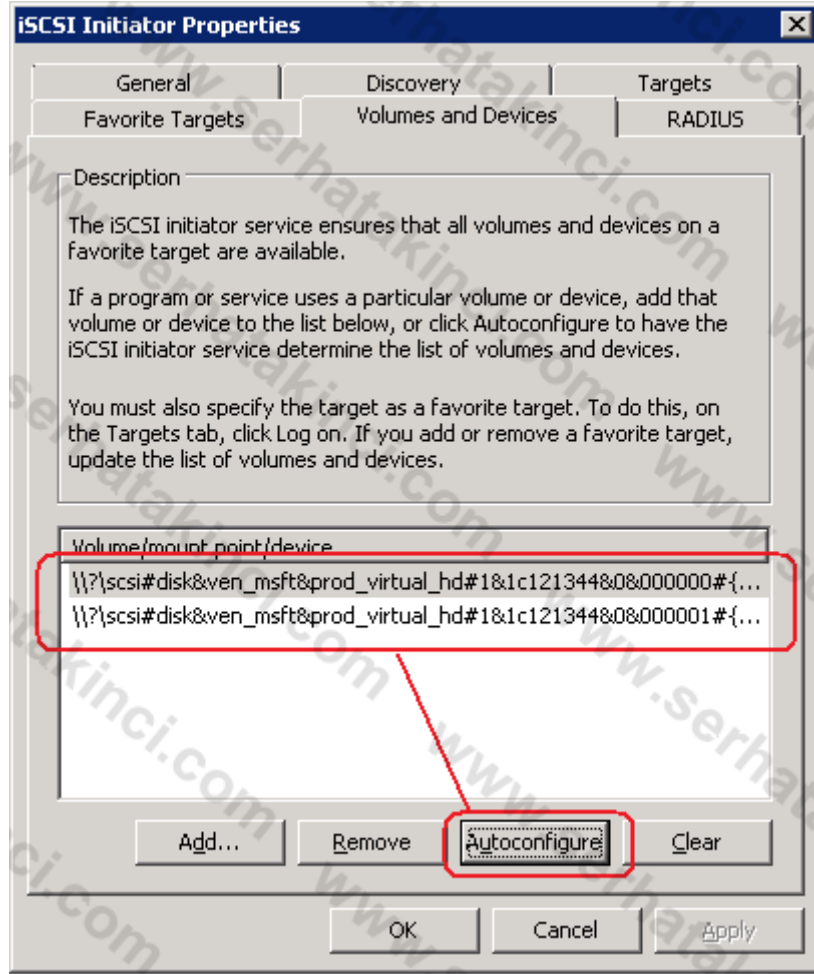


Bu işlemten sonra, **inactive** olan durum bilgisini **connected** olarak görünüyor olmalı.



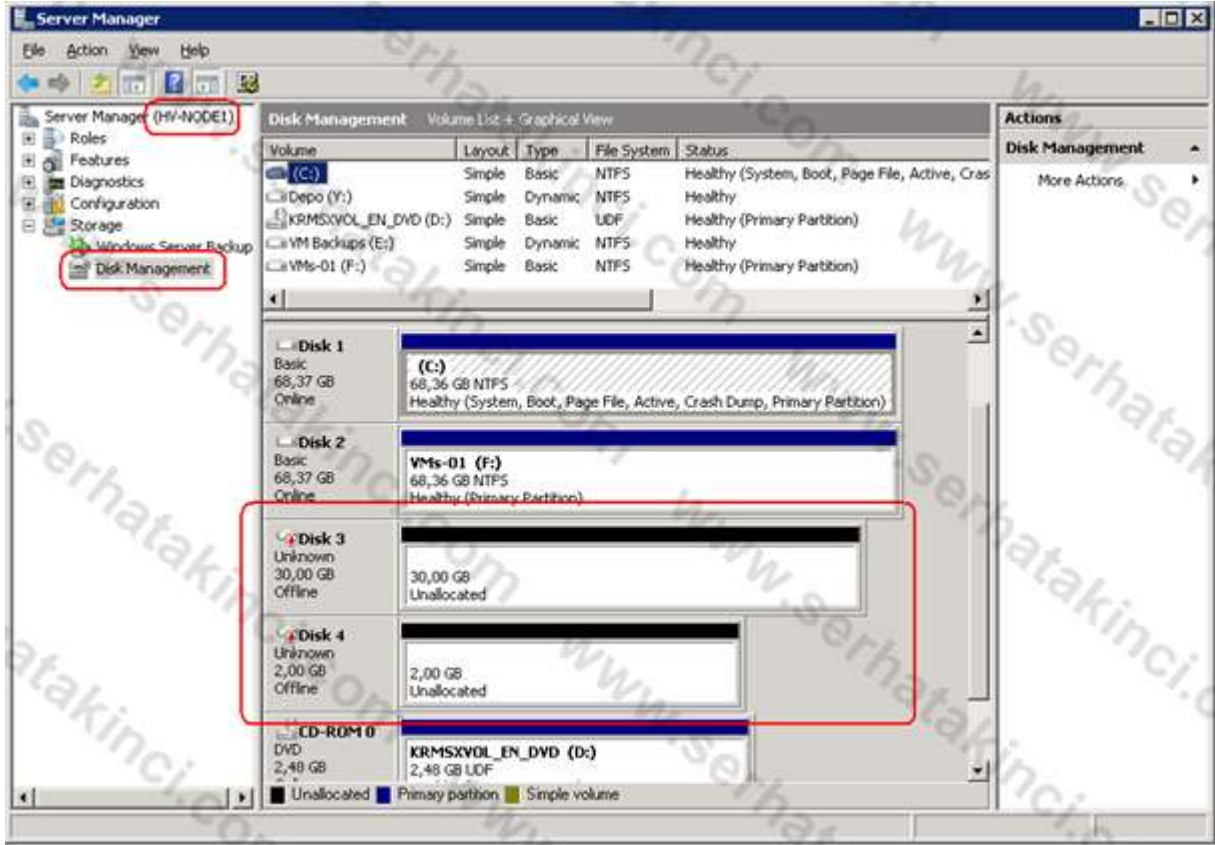
Son olarak **Volumes and Devices** tabına gelip **Autoconfigure** diyoruz ve storage üzerinde bu target için tahsis edilen Virtual Disk'lerin gelmesini sağlıyoruz.

Bu diskler, az önce eklediğimiz disklerdir (iSCSI Storage üzerinde HV-Node1 için disk (virtual disk) eklemek bölümünde).



HV-Node1 için disk ekleme işlemi bitti.

Artık Server üzerinde **Disk Management** bölümünden baktığımızda eklediğimiz iki diski görebiliriz.



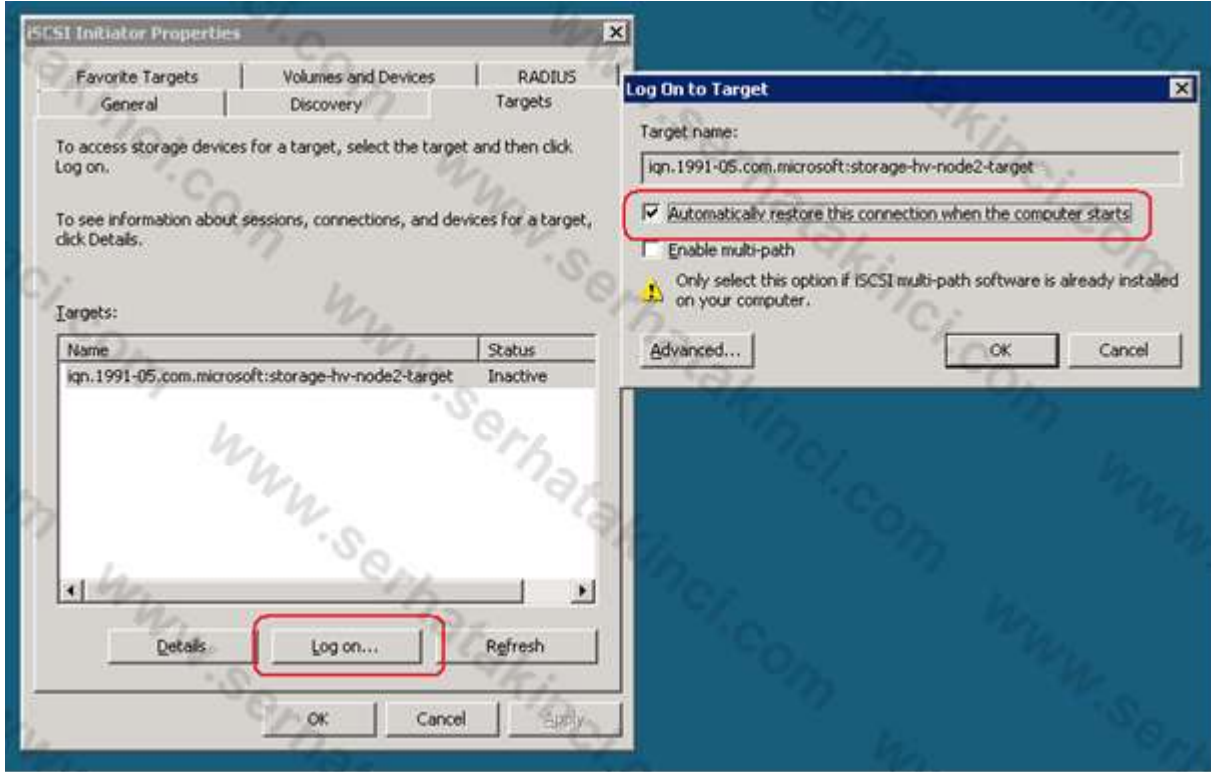
Birisi 2GB, diğeri ise 30GB boyutunda. Bunlar bizim storage üzerinde yaratmış olduğumuz Virtual Diskler. Henüz üzerlerinde partition ve dosya sistemi yok. Ayrıca offline durumdalar.

Aynı işlemleri **HV-Node2** için de yapalım.

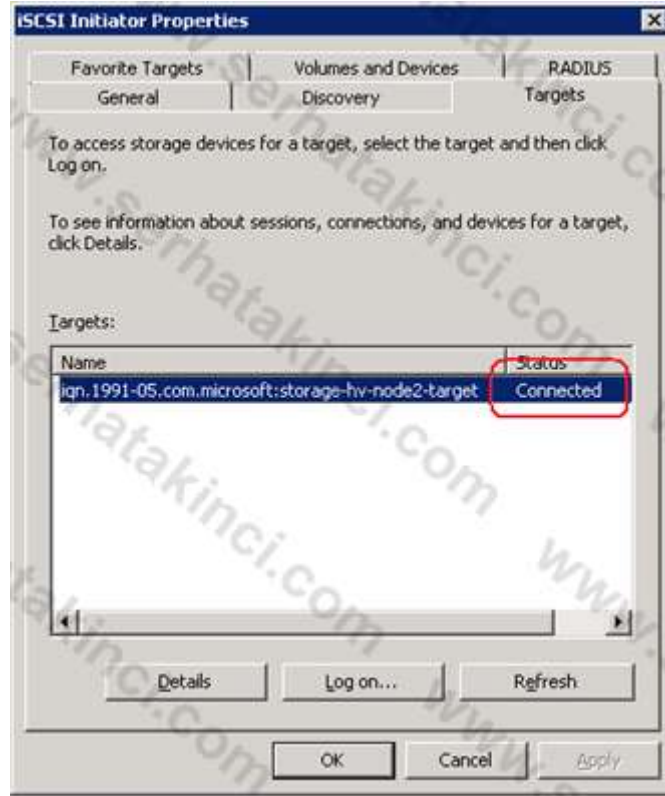
## 7.6. HV-Node2 üzerinde Disk (virtual disk)'i Görmek

**HV-Node2** üzerinde **iSCSI initiator**'ı açıyoruz ve **Targets** tabına geçip **Refresh** diyoruz.

Target listeye geldikten sonra seçerek **Log On...** diyoruz ve **Automatically restore this connection when the computer starts** kutucuğunu işaretleyerek tekrar **Ok** diyoruz.



**Ok** dedikten sonra **Connected** bilgisini görüyor olmalıyız.

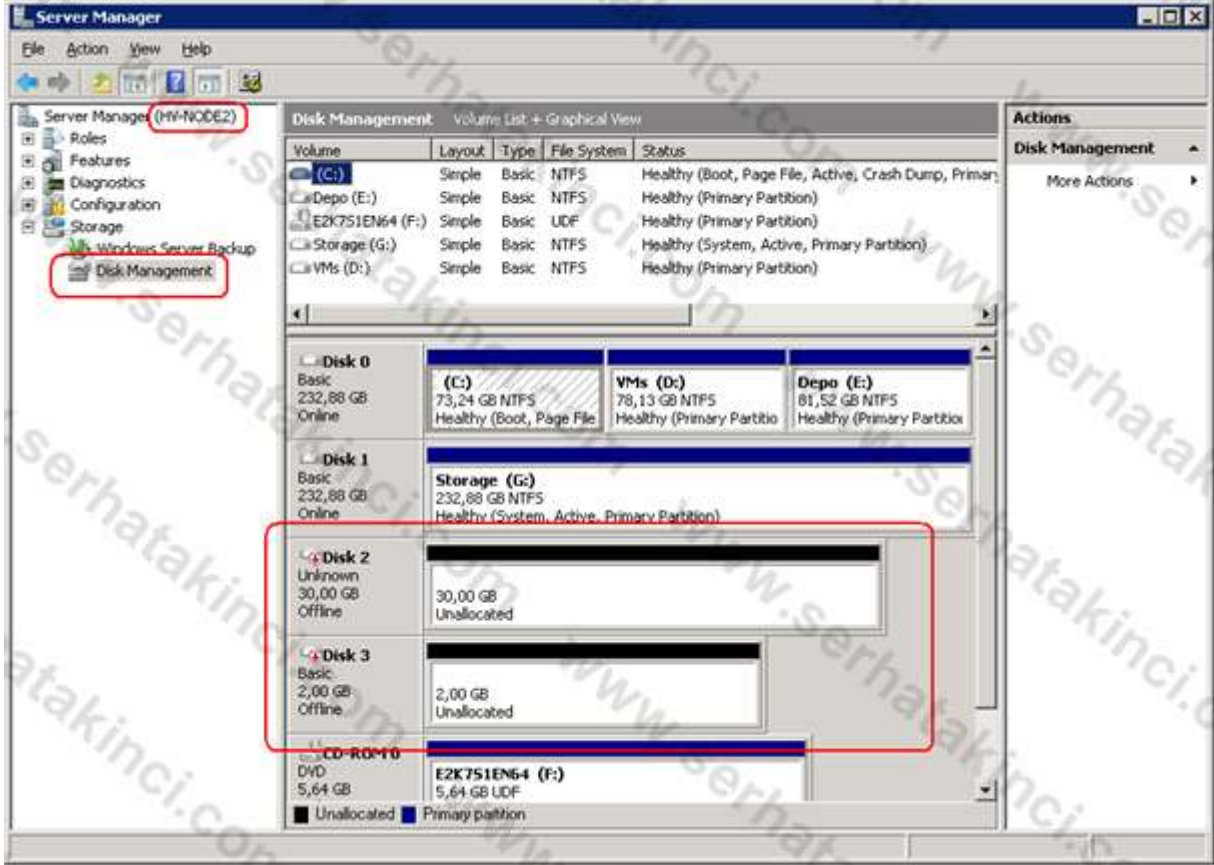


Daha sonra **Volumes and Devices** tabına geçiyoruz ve **Autoconfigure** diyerek storage üzerinde bu target için eklediğimiz disklerin gelmesini sağlıyoruz.



**Ok** dedikten sonra HV-Node2 üzerindeki disk ekleme işlemi de bitmiş oluyor.





Yine **Disk Management** bölümünden kontrol ettiğimizde her iki diski de görüyor olmalıyız.

Böylece storage üzerinde duran iki diski, Hyper-v node'ları üzerine eklemiş olduk ve ortak kullanıma sunduk.

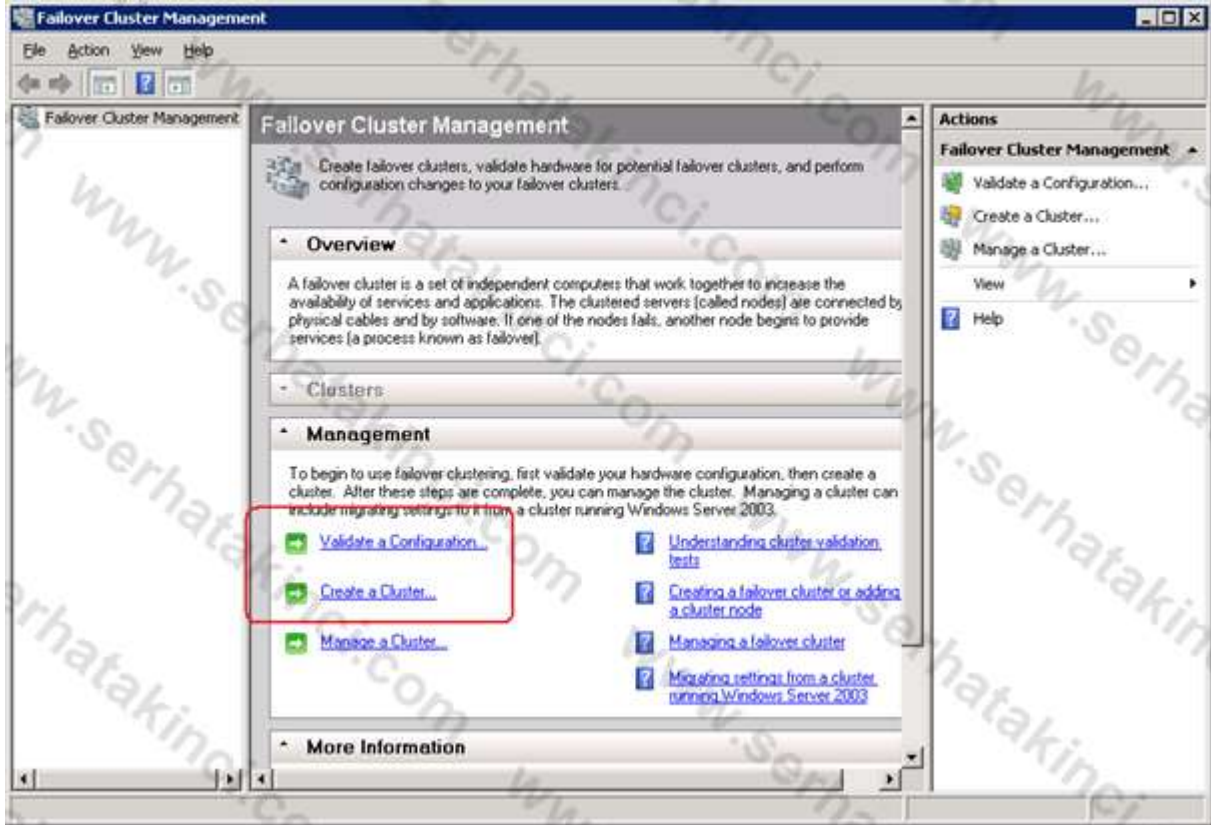
Artık yapımız cluster kurulumu için hazır.

## 8. Failover Cluster Kurulumu

Failover Cluster kurulumunu (konfigürasyonunu) tek bir node üzerinde yapacağız ve bu yapılandırma her iki node'a da etki edecek.

Ben kurulum için ilk node'u seçiyorum (özel bir nedeni yok). HV-Node1 üzerinde **Failover Cluster Management** konsolunu açıyoruz.

### 8.1. Cluster Kurulumu Öncesi Kontrol



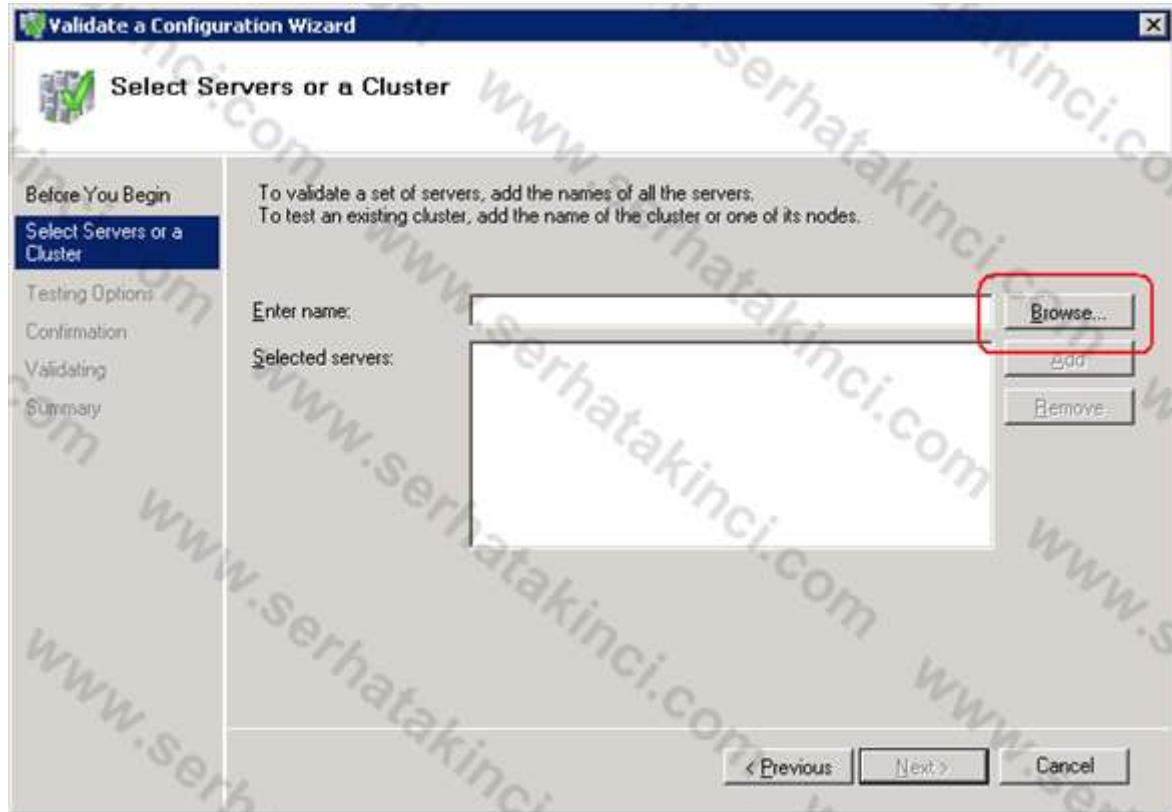
Windows Server 2008'deki bu konsol ve üzerindeki wizard'lar sayesinde cluster kurulumu gerçekten çok basit bir hal almış.

Cluster yaratmadan önce **Validate a Configuration...** wizard yardımı ile cluster için kullanacağımız Nodelar üzerinde kapsamlı bir kontrol gerçekleştirebiliyoruz. Bu kontrol ile Cluster için gerekli olan ihtiyaçlar her iki Node üzerinde kontrol ediliyor ve eksik bir şeyler varsa rapor olarak bize bildiriliyor. Bizde bu sorunlar için önceden önlem alıyor ve gerekli konfigürasyonları yapıyoruz. Böylece hata yapma şansımız minimuma inmiş oluyor.

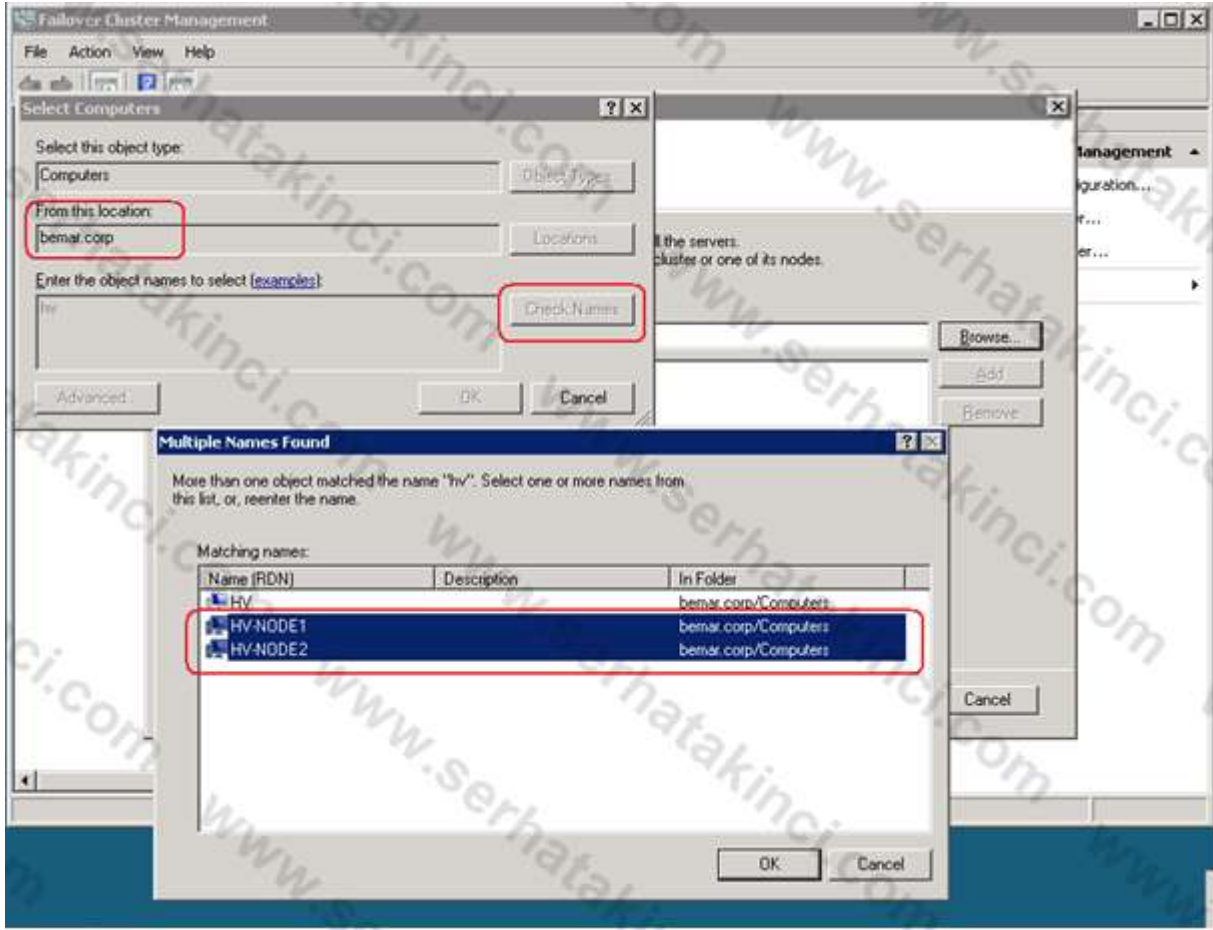
Örnek olması açısından **Validate a Configuration...** wizard'ı çalıştırıyorum.



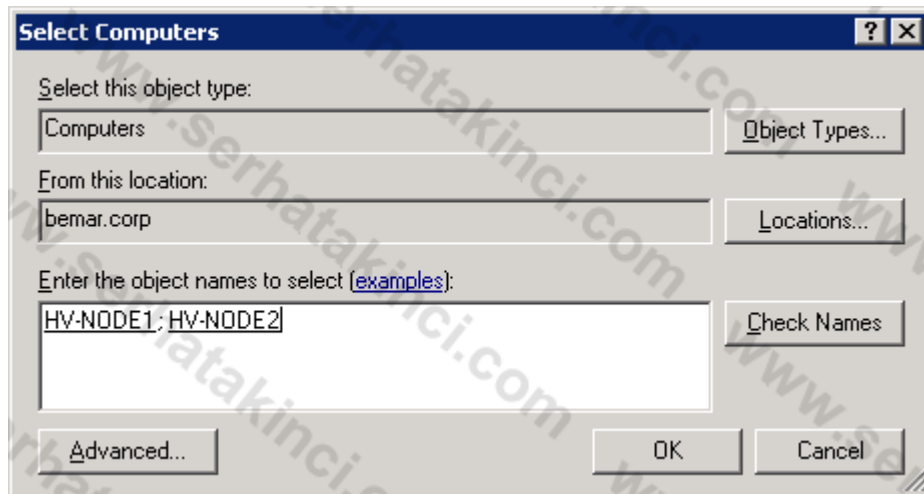
**Browse** butonu ile kontrol yapacağımız yani cluster üyesi yapmayı düşündüğümüz nodeları ekliyoruz.



Gördüğünüz gibi bilgiler active directory ortamından geliyor ve domain'e aldığımız iki node'u görüyoruz.



Her iki node'u aynı anda seçip **Ok** diyoruz.



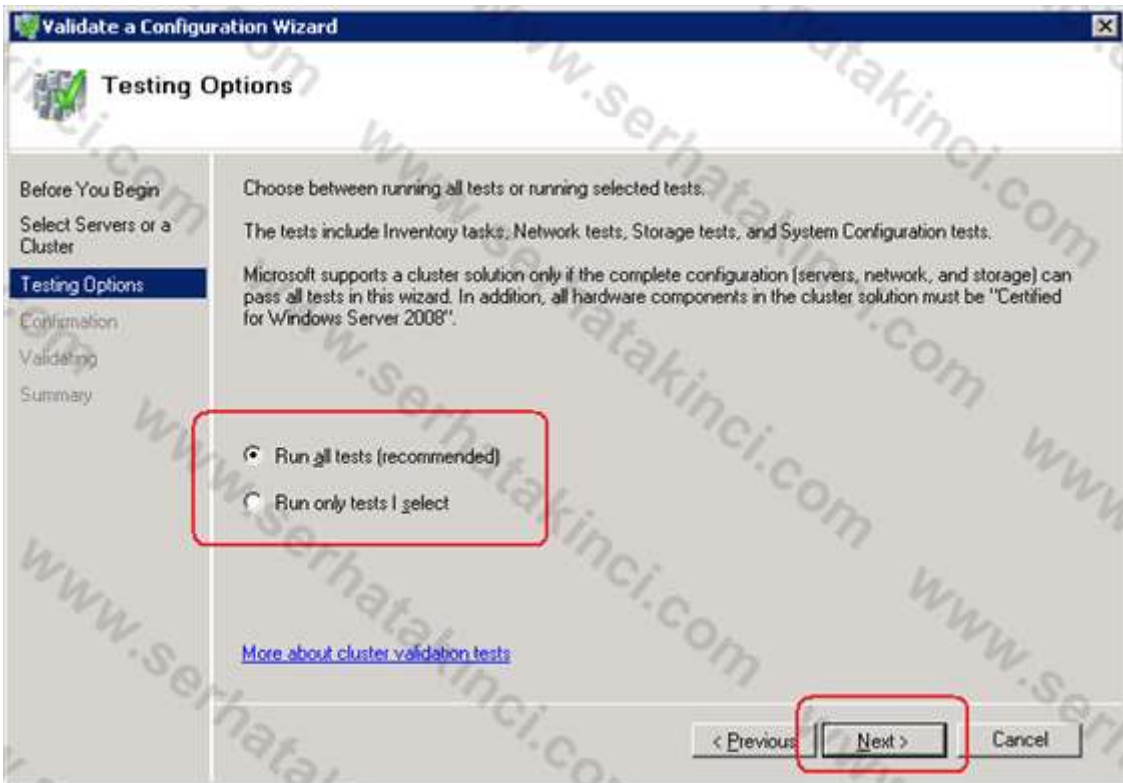
Ve nodelar **selected servers** bölümüne ekleniyor.



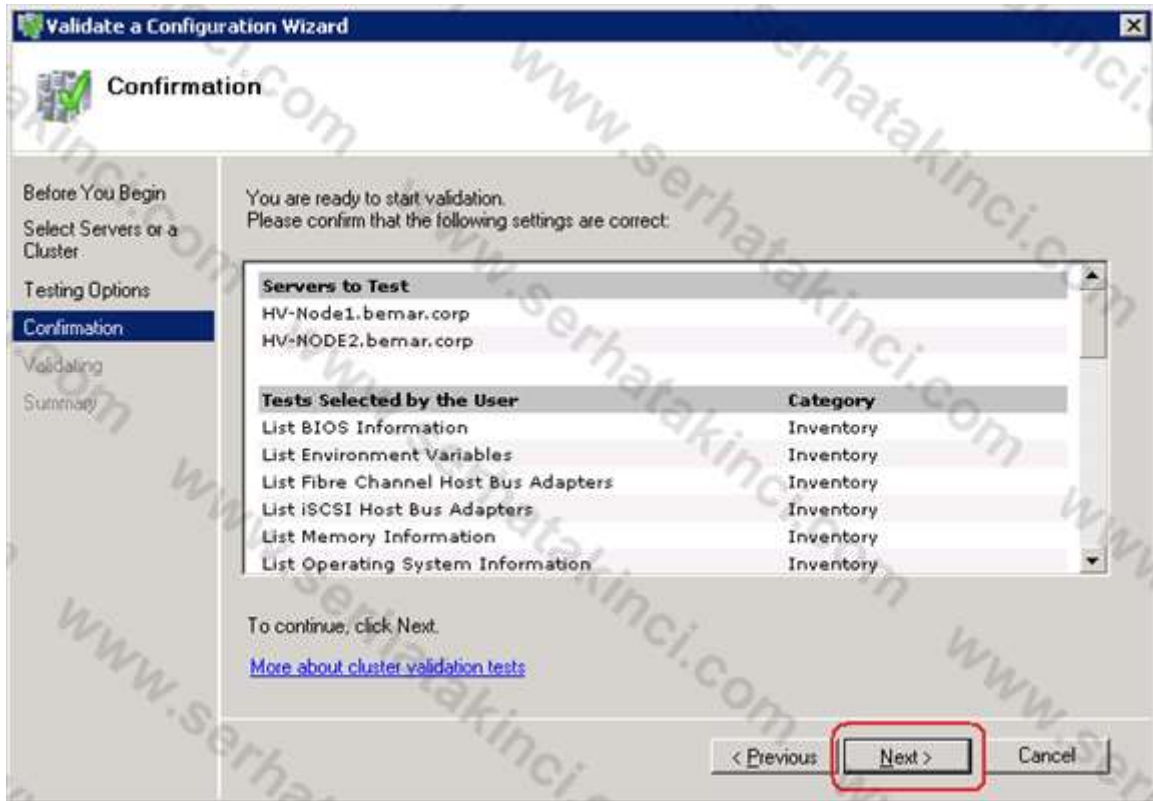


**Next** dedikten sonra gelen **Testing Options** penceresinde **Run all tests** diyerek gerekli tüm testlerin yapılmasını sağlayabileceğimiz gibi, **Run only tests I select** diyerek seçim yapıp, network, storage, os configuration gibi belirli özelliklerin test edilmesini de sağlayabiliyoruz.

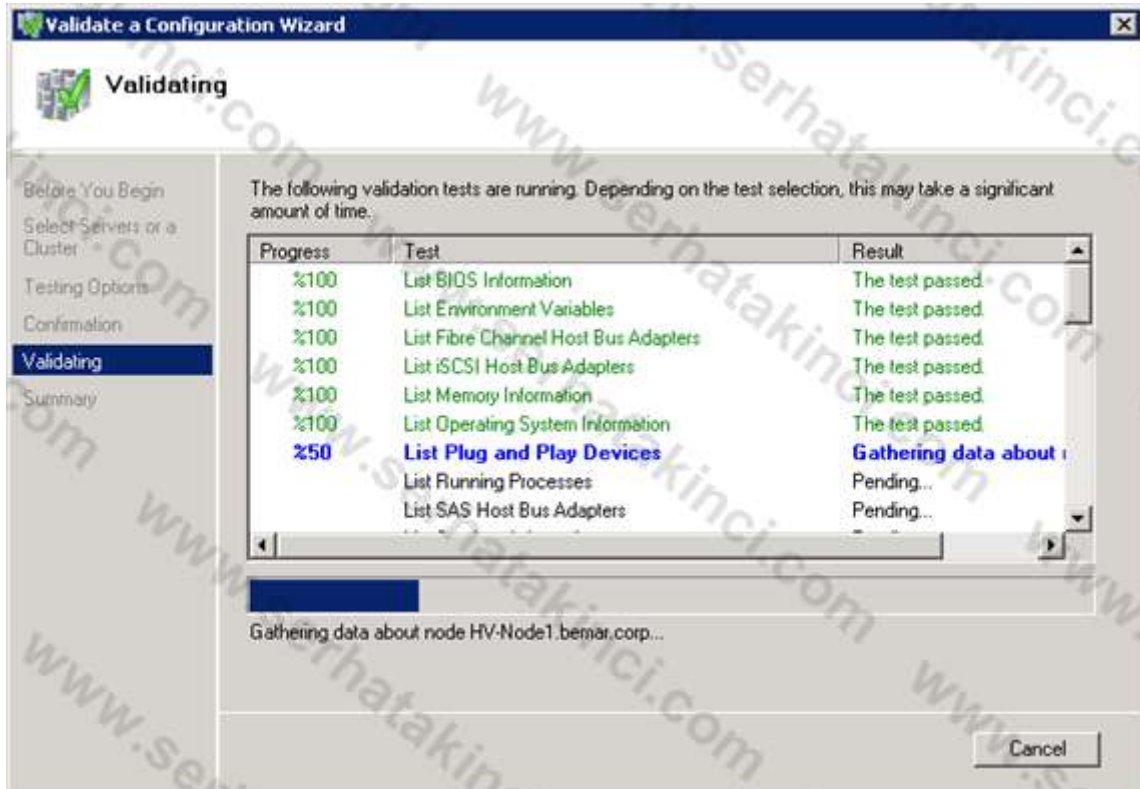
**Run all tests** ile devam edelim.



Next diyerek doğrulamayı başlatıyoruz.



Kontrol işlemi biraz zaman alabilir. Süreci izleyebiliyorsunuz.





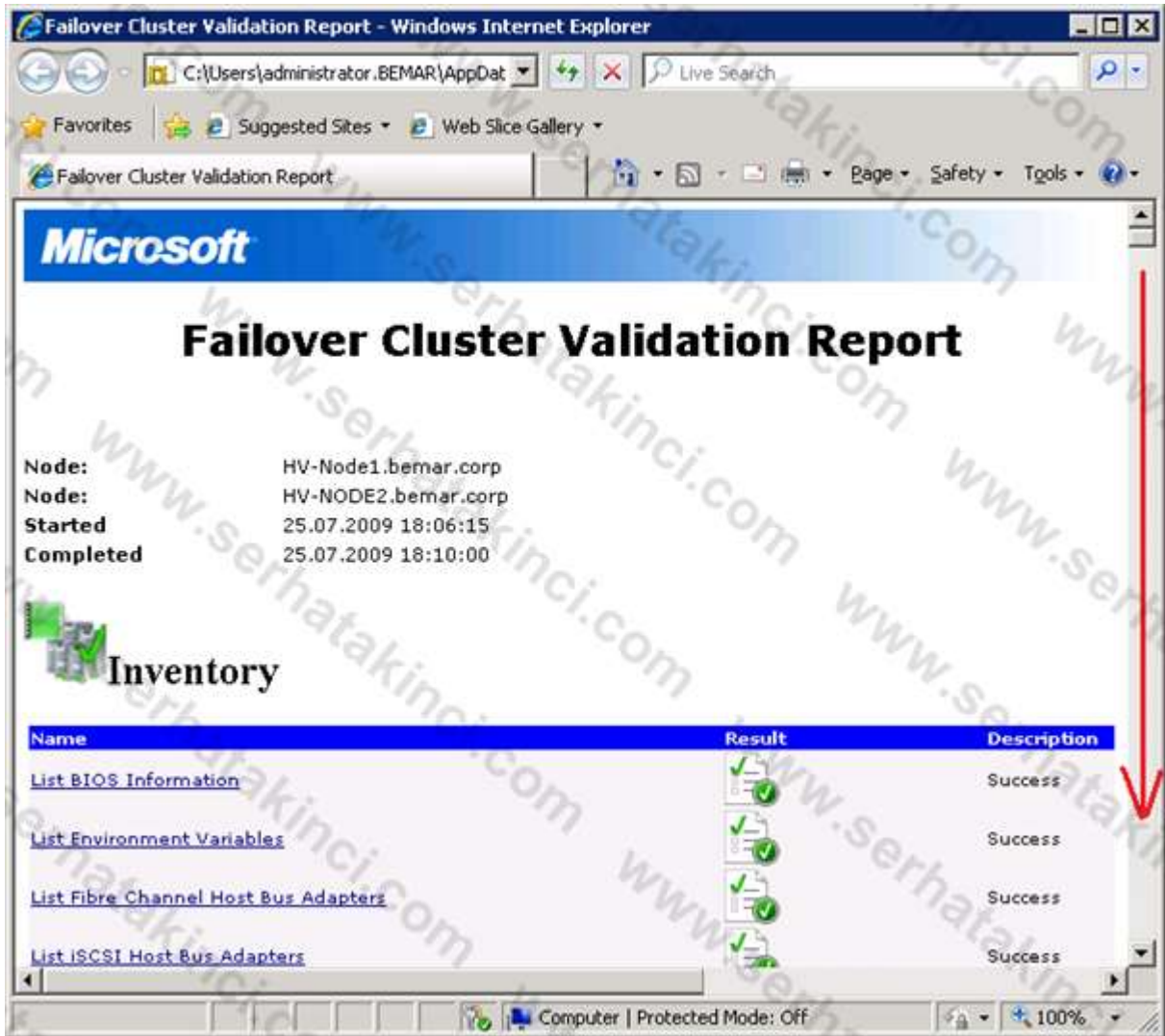
Kontrol tamamlandıktan sonra **successfully** bilgisini görüyoruz. Bu testin başarılı bir şekilde tamamlandığını gösteriyor.



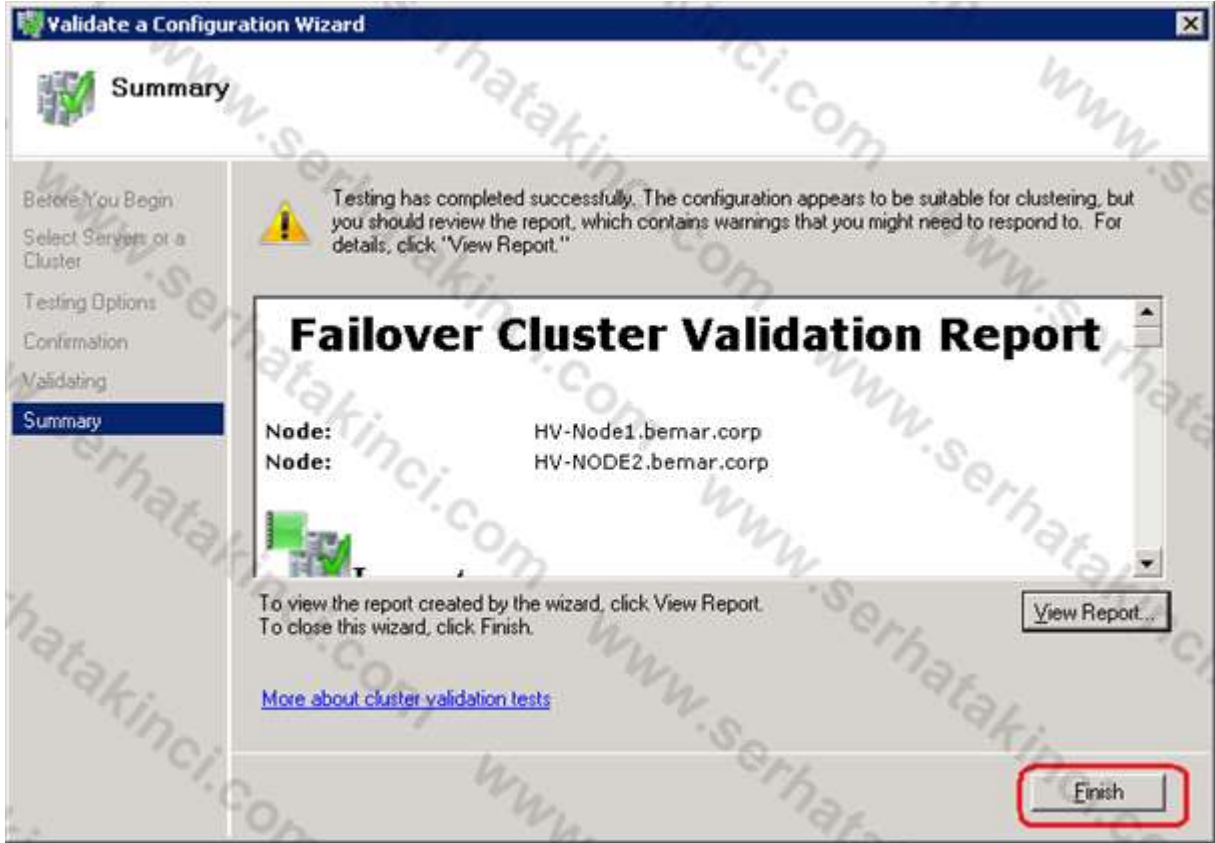
Ama bir uyarı var. Bu uyarının anlamı şu: kontrollerde bazı noktalarda eksikler tespit edildi ama bu noktalar cluster kurulumu için engel değil çünkü bunlar cluster sonrasında da düzeltilebilecek eksikler (ihtiyaç halinde).

**View Report** butonuna basarak raporun detaylarını görebilir, eksiklerin nerede olduğunu listeleyebilirsiniz.

**View Report** tıkladıktan sonra browser üzerinde bir rapor geliyor. Bu sayfada aşağı doğru ilerleyerek bilgileri inceleyebilirsiniz.



**Finish** ile wizard'ı kapatıyoruz.

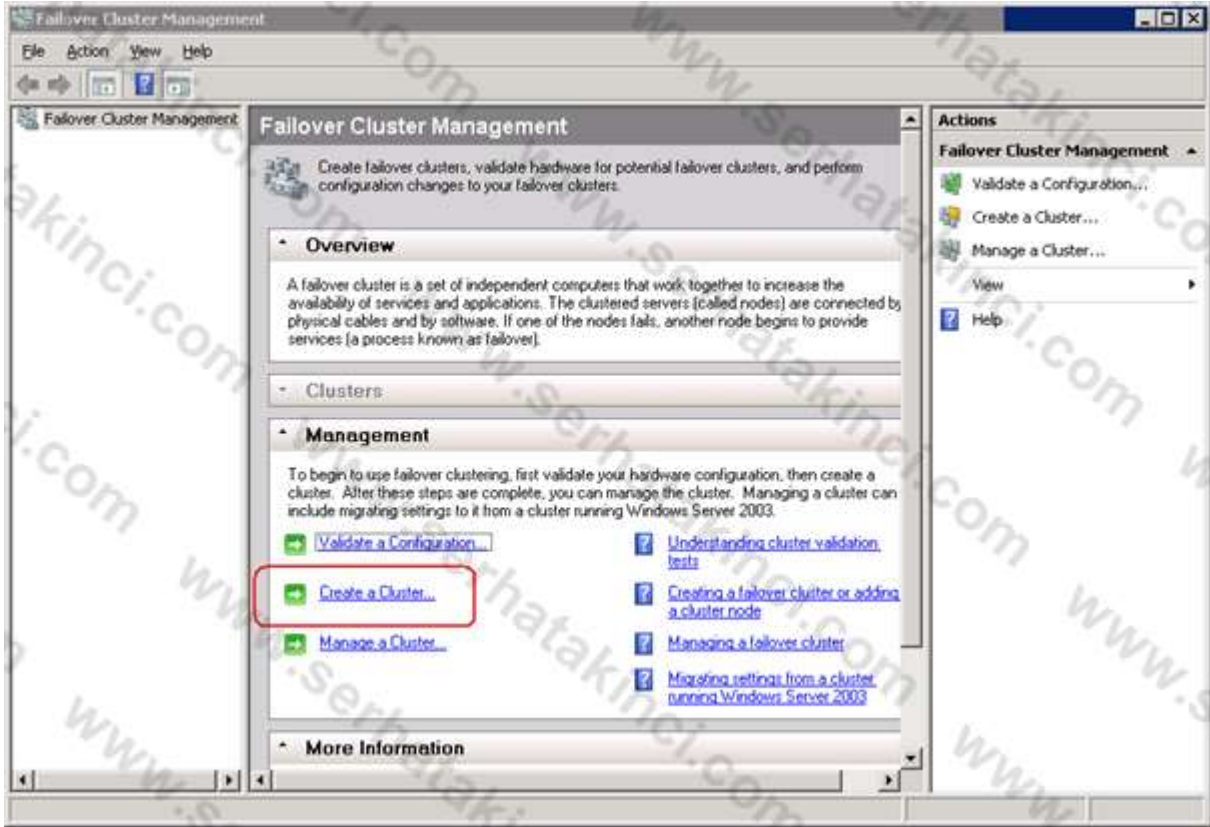


Eğer ki cluster için büyük bir sorun olsaydı succeeded mesajı yerine hatalar görecektik ve bu da cluster öncesi mutlaka düzeltilmesi gereken sorunlar anlamına gelecekti.

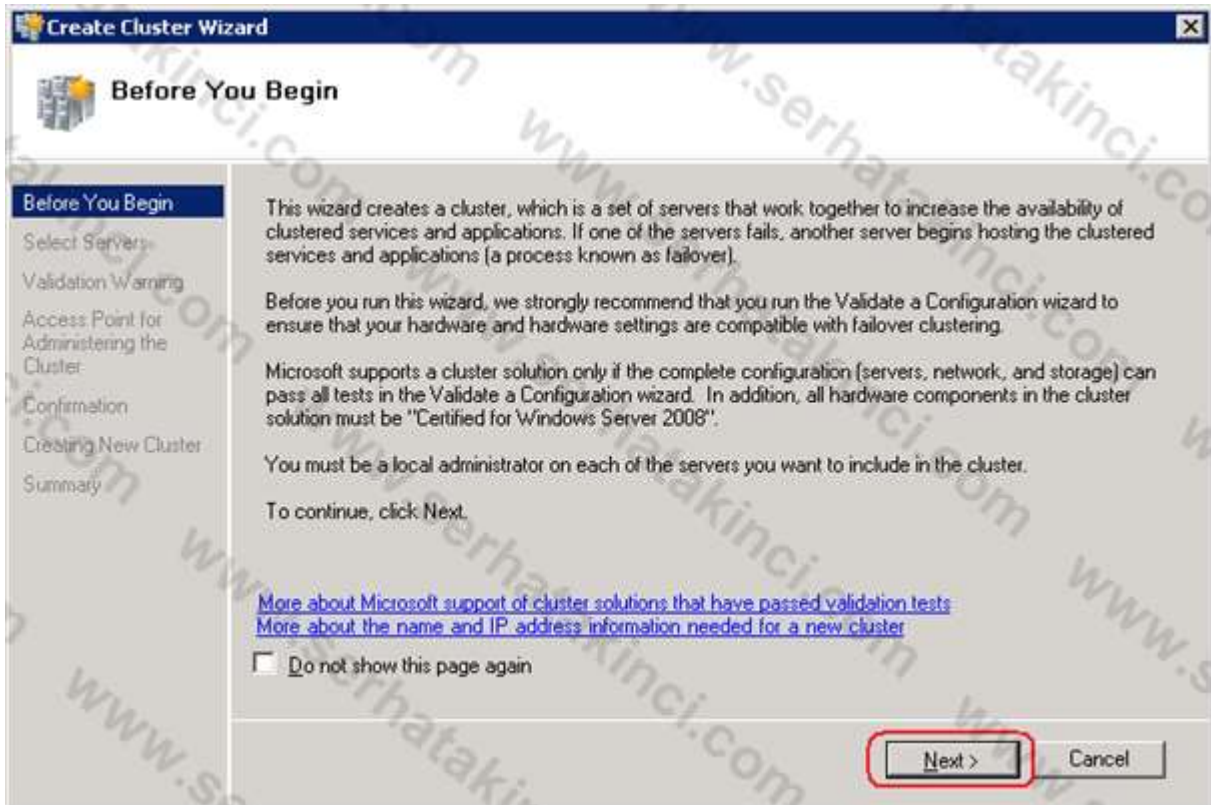
Büyük bir sorun olmadığına göre Cluster'ı oluşturalım.

## 8.2. Cluster Kurulumu

HV-Node1 üzerinde, **Failover Cluster Management** konsolunda **Crete a Cluster** diyerek wizard'ı açıyoruz.



**Next** ile ilerliyoruz.



Nodeları ekleyip devam ediyoruz.





Aşağıdaki Access Point penceresi önemli bir bölüm.

Burada öncelikle cluster için bir isim veriyoruz. Bu isim active directory içerisinde bir computer olarak tanımlanacak (otomatik olarak). Bu nedenle var olmayan ve cluster'ı temsil etmesini istediğiniz bir isim vermelisiniz.

Yine aşağıdaki pencerede seçmiş olduğumuz network üzerinde uygun bir IP veriyoruz. Bu ip adresi de nodelardan farklı ve network üzerinde var olmayan bir IP olmalı çünkü cluster'ı temsil eden IP bu olacak.

İkinci network üzerinde cluster hizmeti vermeyeceğiz, onun için seçimini kaldırıyorum. 3ncü network ise özel bir durumdan dolayı bende gözüküyor.

**Next** ile ilerliyoruz.

**Create Cluster Wizard**

**Access Point for Administering the Cluster**

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

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

Cluster 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.

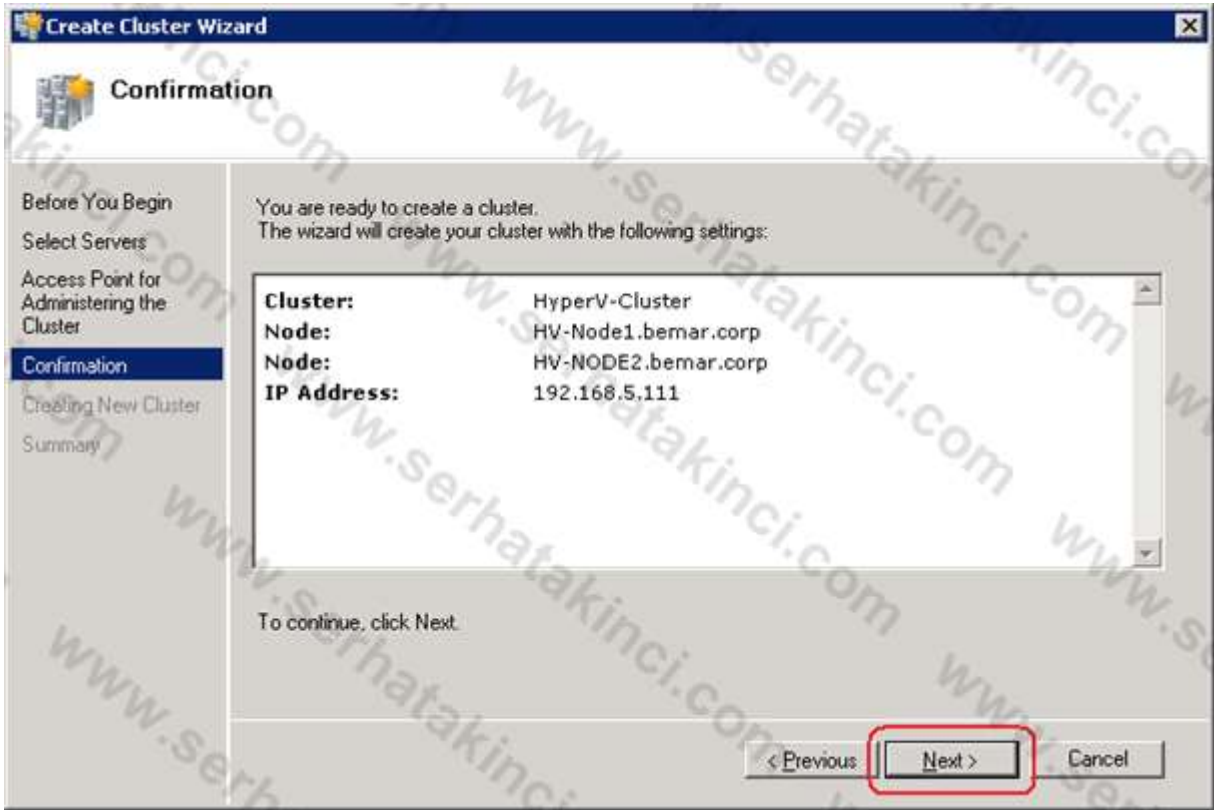
	Networks	Address
<input checked="" type="checkbox"/>	192.168.5.0/24	192.168.5.111
<input type="checkbox"/>	11.0.0.0/8	Click here to type an address

[More about the administrative Access Point for a cluster](#)

< Previous **Next >** Cancel



**Confirmation** penceresinde **Next** diyerek kurulumu başlatıyoruz.



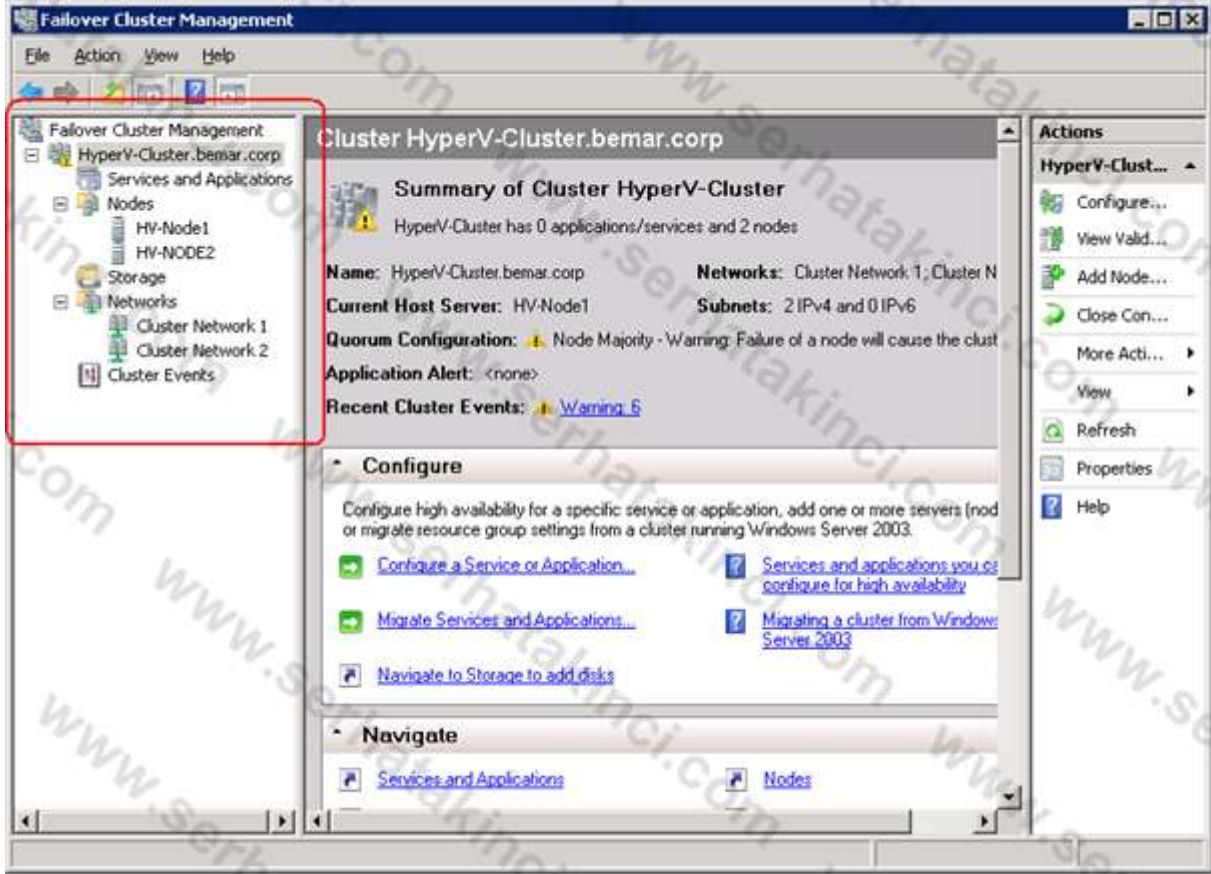
Kurulum fazla uzun sürmeyecektir.



Kurulum tamamlandıktan sonra **Successfully** mesajını görüyoruz ve **Finish** ile wizard'ı kapatıyoruz.

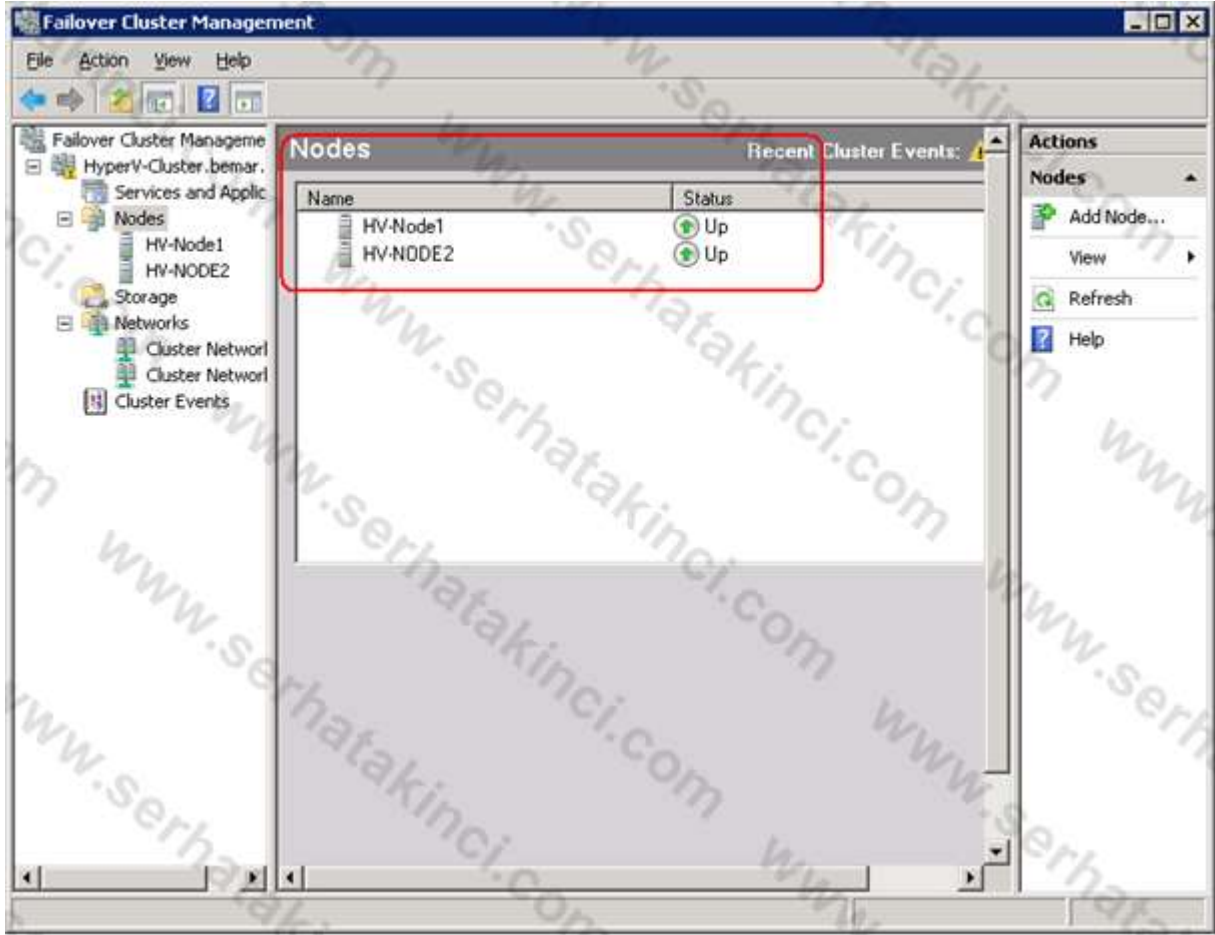


**Failover Cluster Management** konsoluna dönüyoruz ve eğer bir sorun yoksa her iki Node un da eklenmiş olduğunu görebiliyoruz.

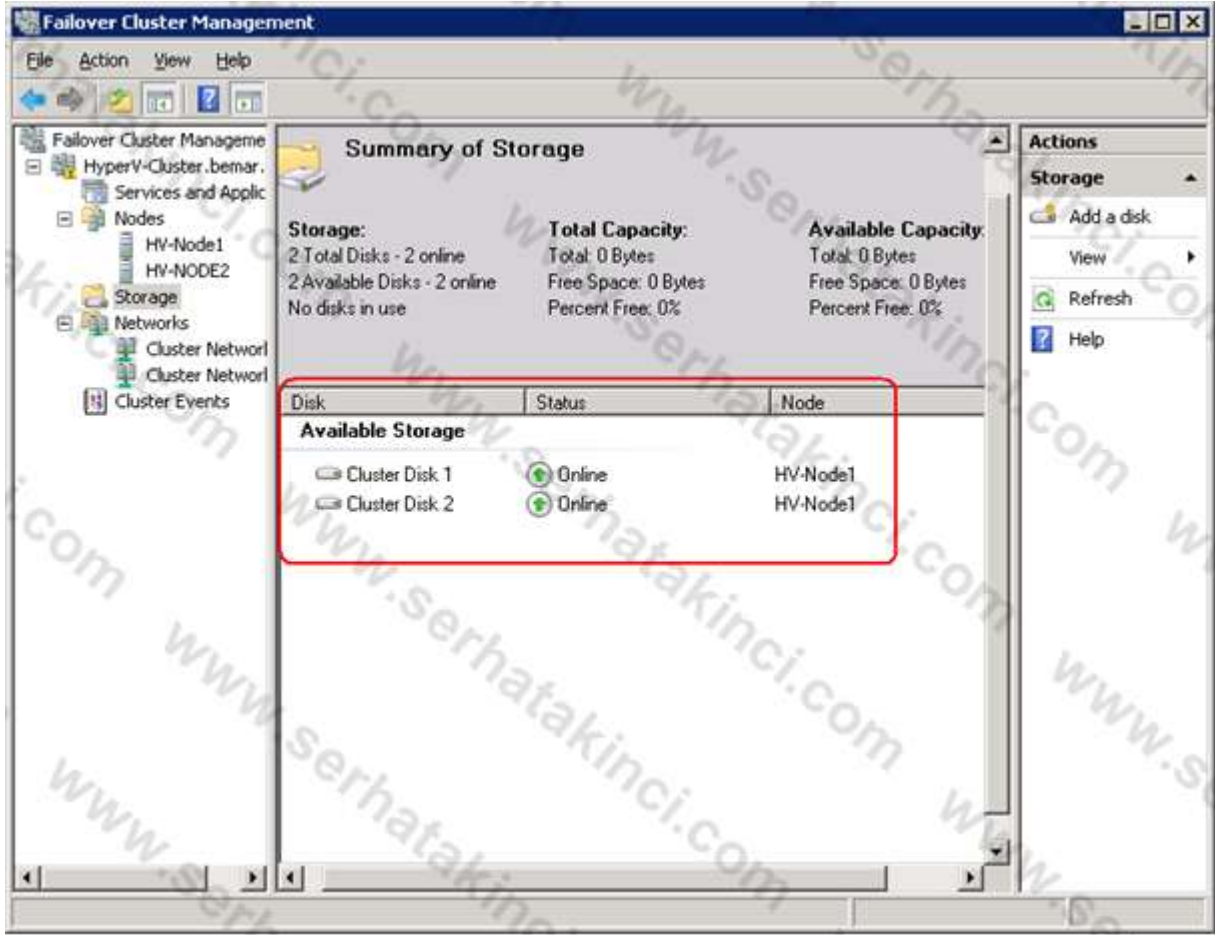


Artık iki fizksel sunucumuz aynı cluster'ın üyesi olarak çalışıyor ancak henüz herhangi bir servis yada uygulama ekli değil.

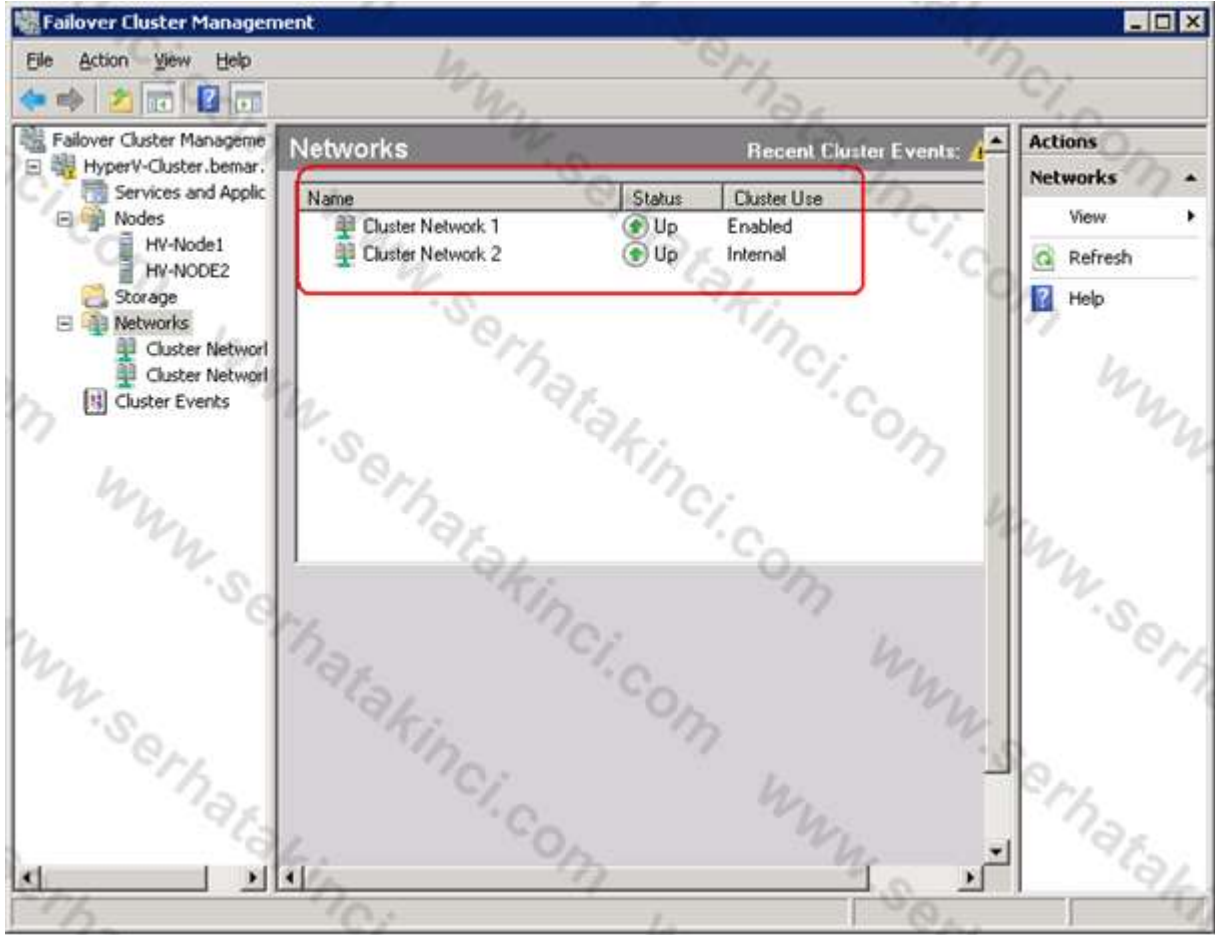
**Nodes** bölümüne gelerseniz iki node'un da **up** durumda olduğunu görebilirsiniz.



**Storage** bölümünde ise Node'lar üzerine **iSCSI initiator** ile eklediğimiz ve storage üzerinde duran iki adet disk görebiliyoruz. Bu diskler şu an clustered ve online durumda.



**Networks** bölümünde ise node'lar üzerindeki cluster networkler yer alıyor. Burada 3 network yer alması gerekiyor ama dediğim gibi benim sistemimdeki özel bir durum nedeniyle ikisi listeleniyor.



Biz cluster yapılandırmasını HV-Node1 üzerinde yaptık. Aynı işlemi ikinci node üzerinde yapmaya gerek yok. İkinci yani HV-Node2 üzerinde de Failover Cluster Management konsolunu açarsanız, nodeların ve varsa diğer cluster kaynaklarının aynı şekilde orada da listelendiğini görebilirsiniz.

### 8.3. Quorum Disk Konfigürasyonu

Quorum her cluster yapısında bulunmaz. Sadece failover-server olarak geçen clusterlarda bulunur. Temel amacı cluster bilgisini tutmak ve bu bilgiyi o an cluster kaynaklarının sahibi olan (doğal olarak quorum'un da sahibi) nodeun kullanımına sunmaktır.

Hyper-V üzerinde çalışan sanal makineler için oluşturduğumuz cluster yapısı da bir failover cluster olduğu için quorum ihtiyacı vardır.

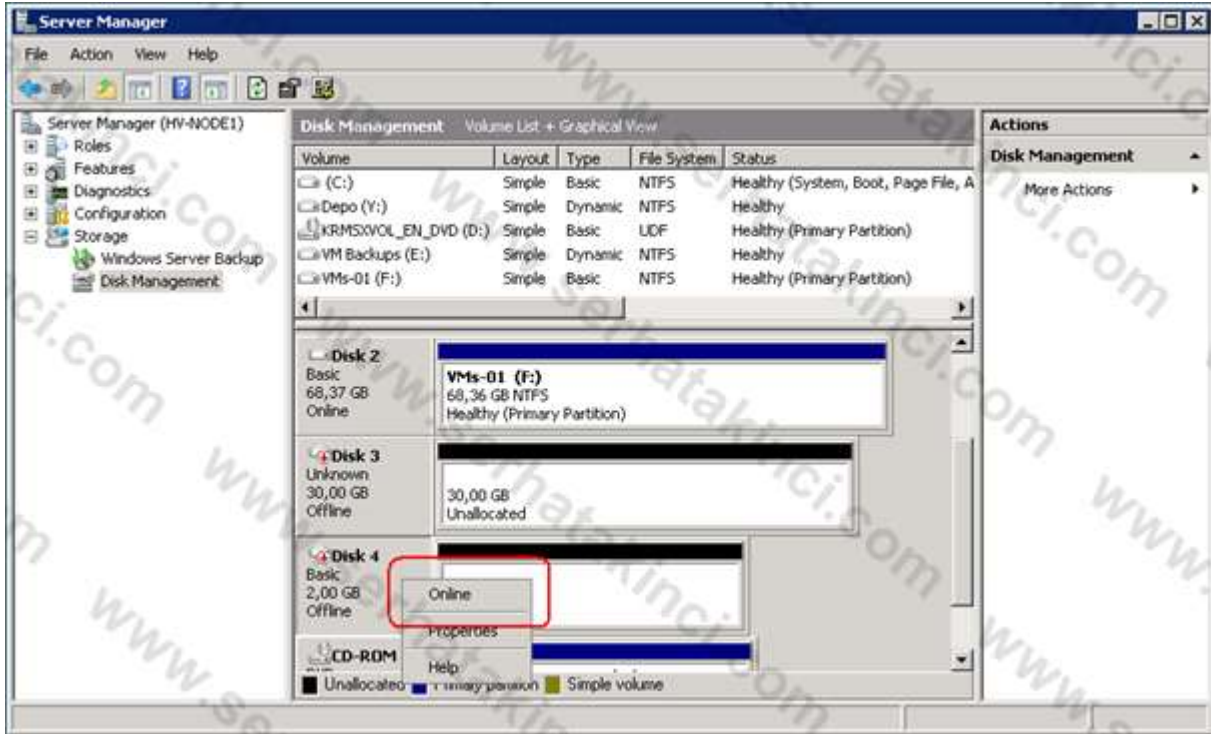
Hatırlarsanız storage üzerinde quorum için kullanılmak üzere 2GB boyutlu bir Virtual Disk oluşturup bunu nodelara atamıştık.

Cluster kurulumunu HV-Node1 üzerinde yaptığımız ve clustered disklerin şu anki sahibi HV-Node1 olduğu için Quorum yapılandırmasına da HV-Node1 üzerinde devam ediyorum.

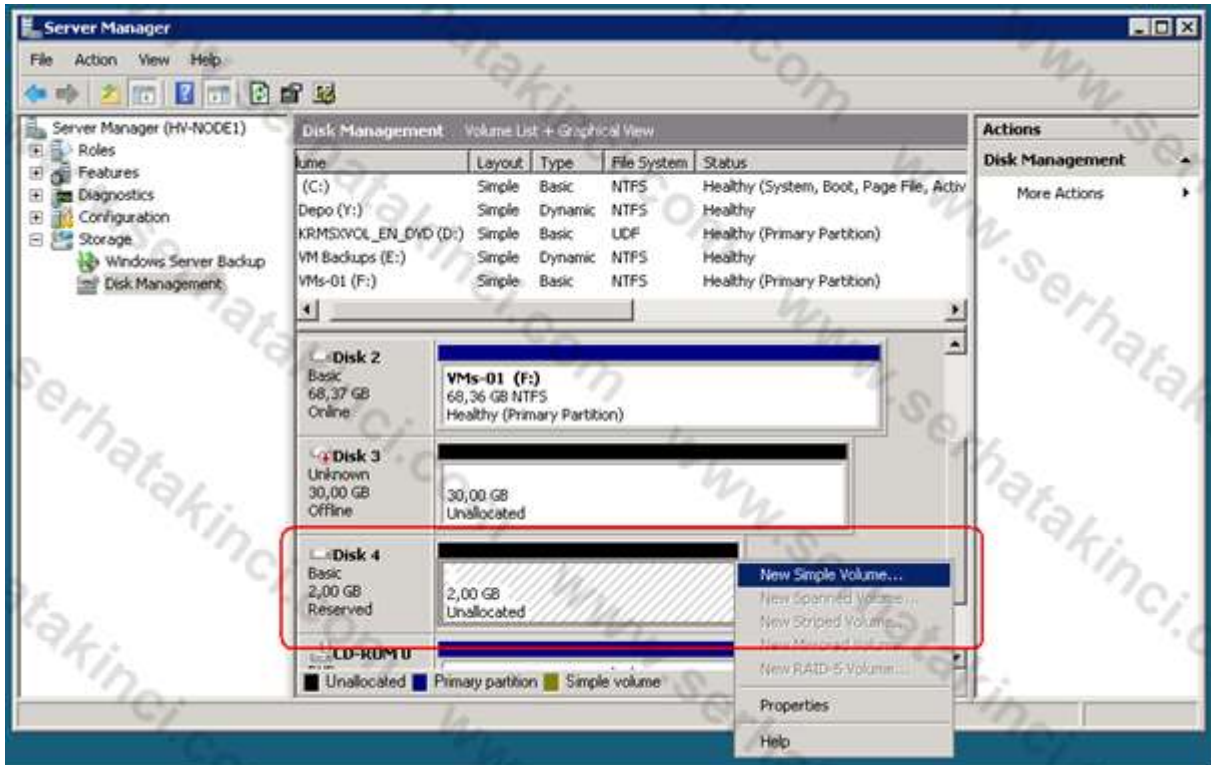


Önce HV-Node1 üzerinde NTFS formatlı bir partition yaratmalıyız. Bunun için **Disk Management**'ı açıyoruz.

Öncelikle ilgili diski online yapıyoruz.



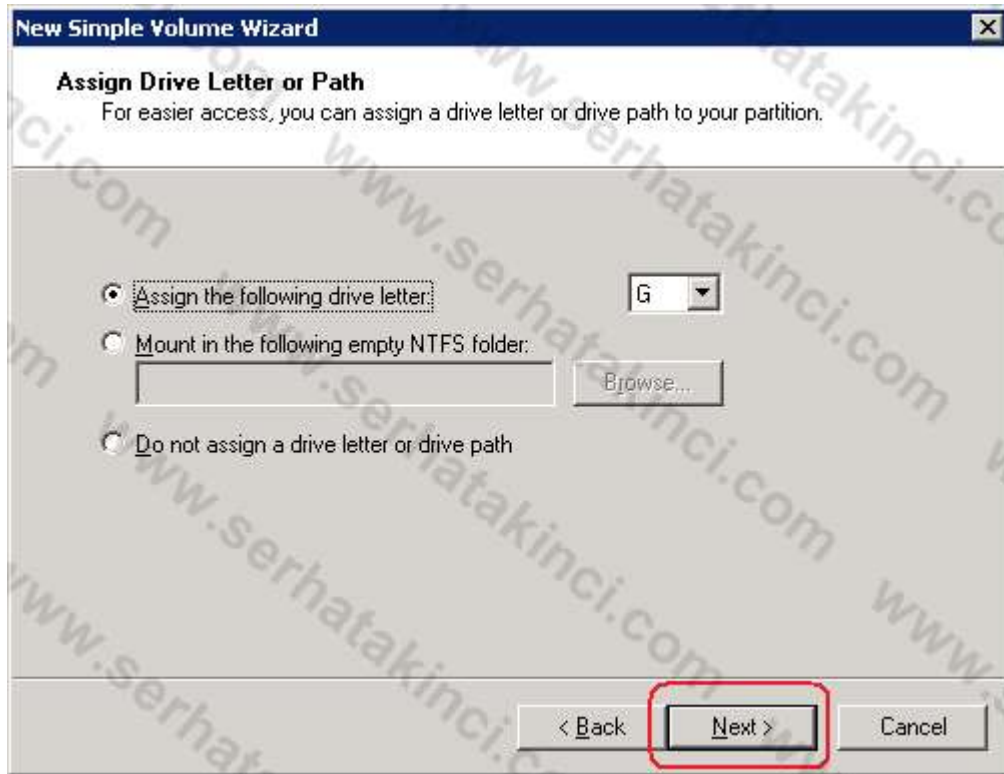
**New Simple Volume** diyerek yeni bir volume yaratmak için wizardı başlatıyoruz.



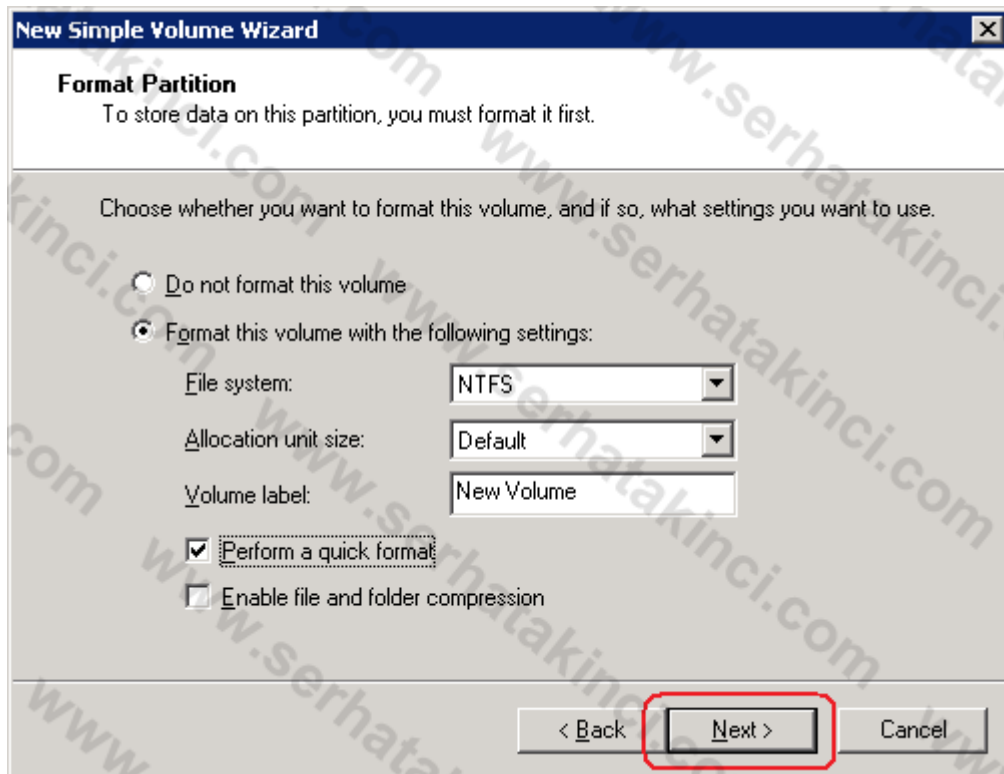


Volume size'ı deęiřtirmeden devam ediyorum.

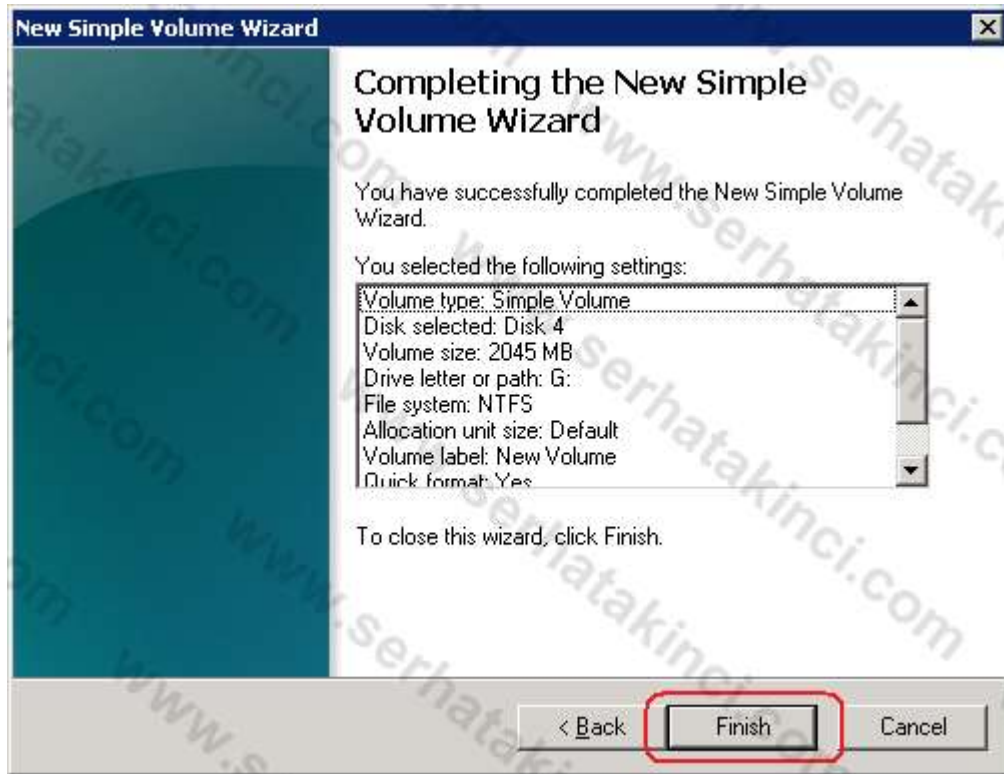




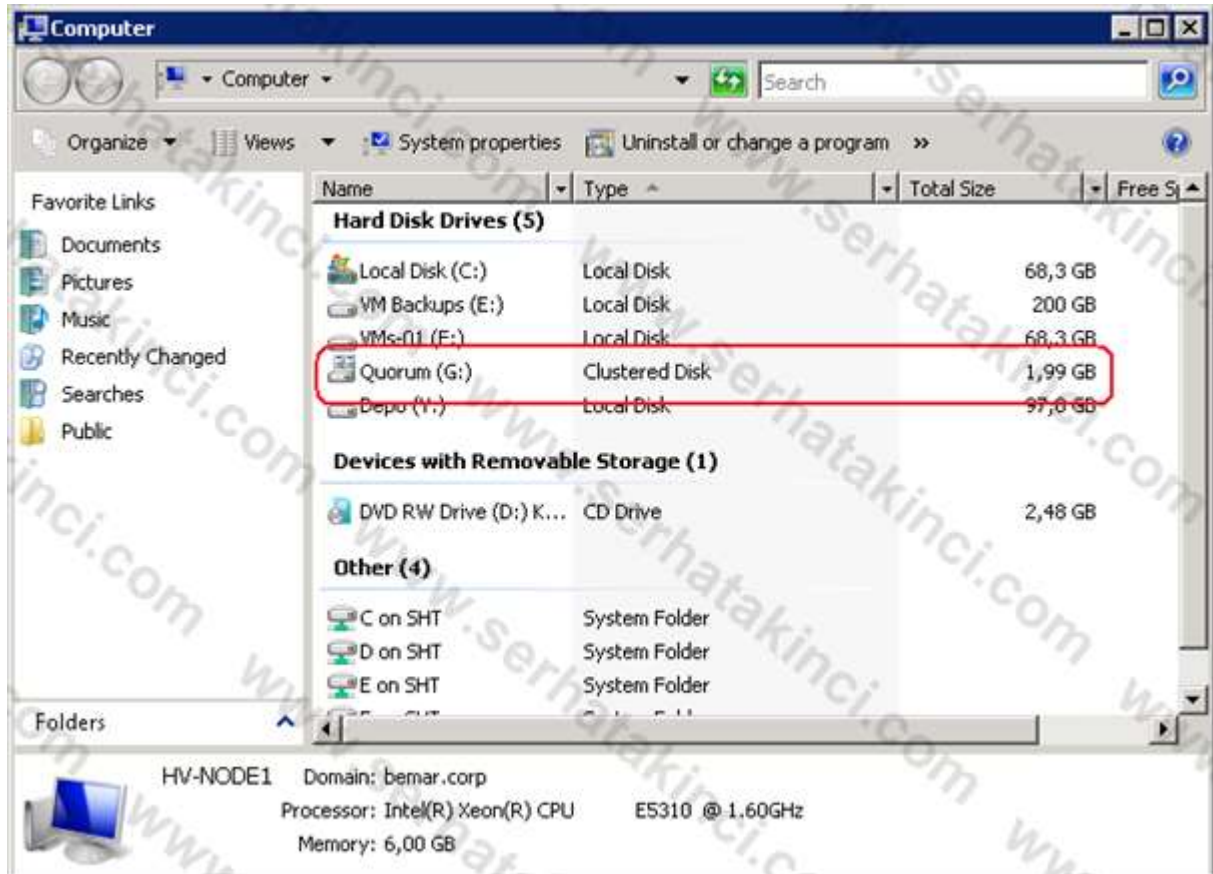
NTFS olarak formatlıyoruz.



Finish ile bitiriyoruz.



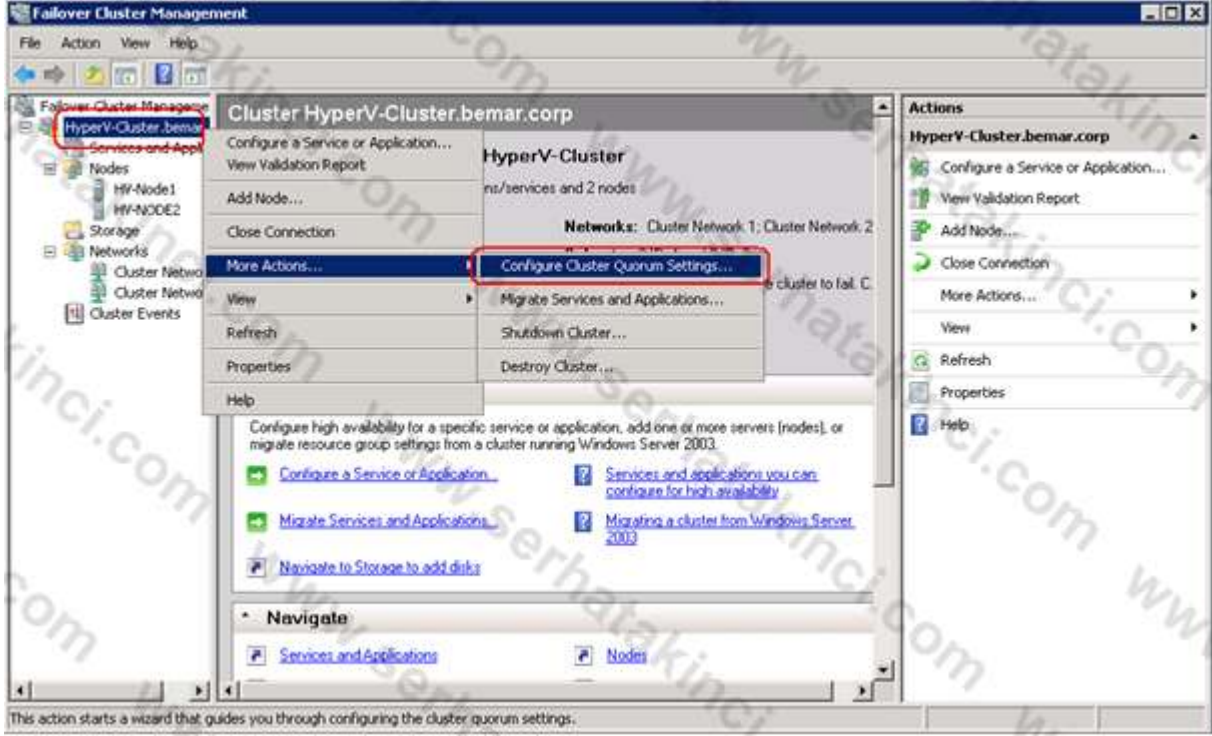
Bu işlem sonrası volume **Computer** altına geliyor ve dikkat ederseniz **Type** olarak **Clustered Disk** görünüyor. İsmi ise ben **quorum** olarak set ettim.





Aynı disk HV-Node2 üzerinde Computer altında görünmez çünkü aynı anda sadece bir node cluster kaynaklarına (disk, uygulama/servis vs..) sahip olabilir.

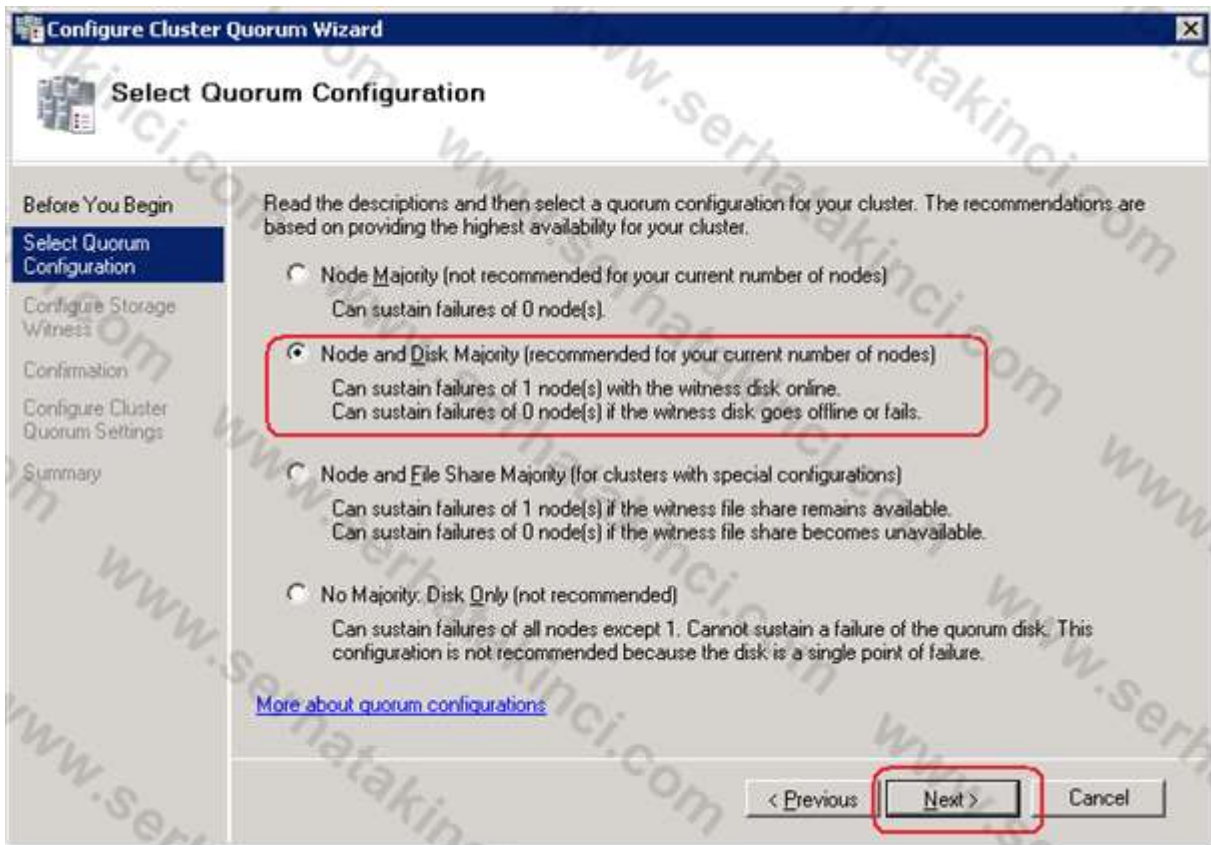
Quorum konfigürasyonu için yine **HV-Node1** üzerinde **Failover Cluster Management** konsolunu açıyoruz ve **Configure Cluster Quorum Settings** diyoruz.



**Next** ile devam ediyoruz.



Node and Disk Majority seçiyoruz.



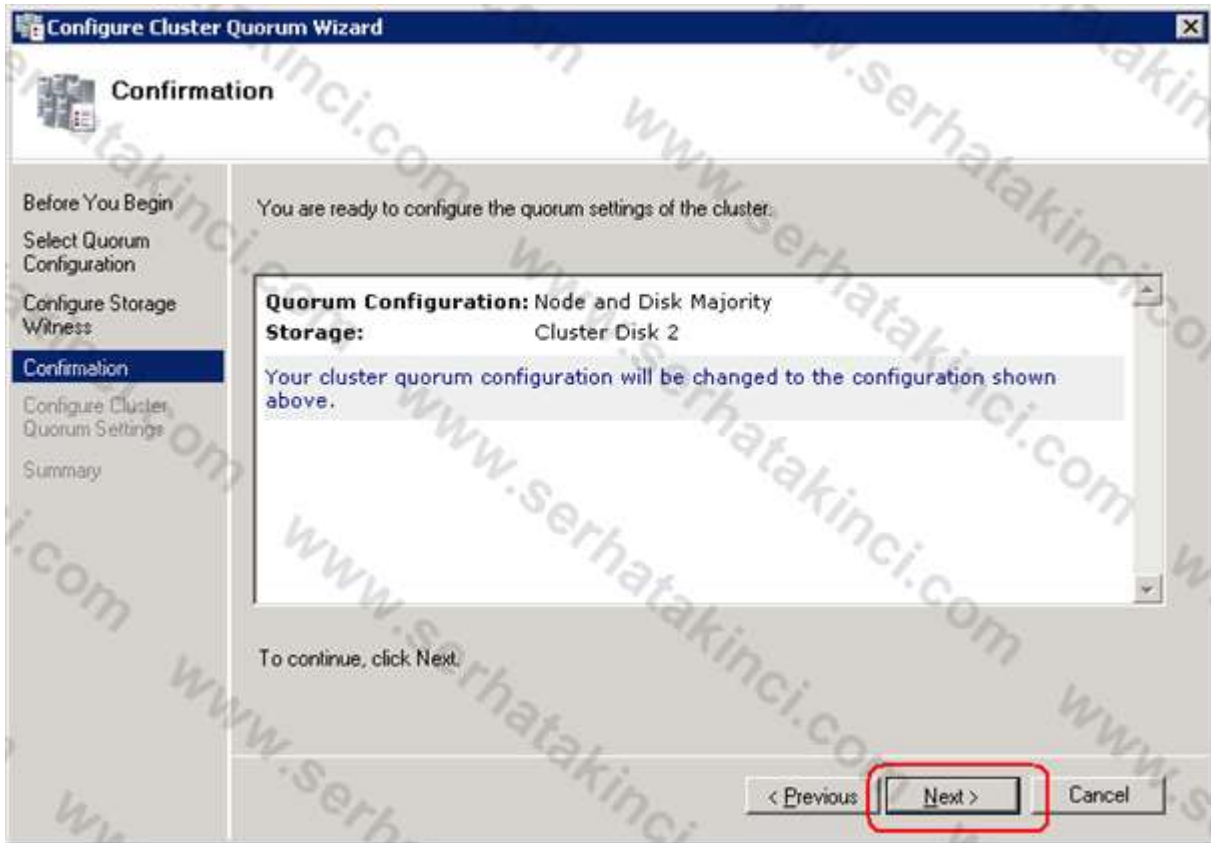


Quorum bilgisinin hangi volume üzerinde duracağını belirtiyoruz. Aşağıdaki pencerede clustered diskler üzerindeki volume'lar listelenir. Şu an için iki clustered diskimiz var ancak sadece birisi üzerinde volume oluşturduk bu nedenle sadece o listeleniyor.

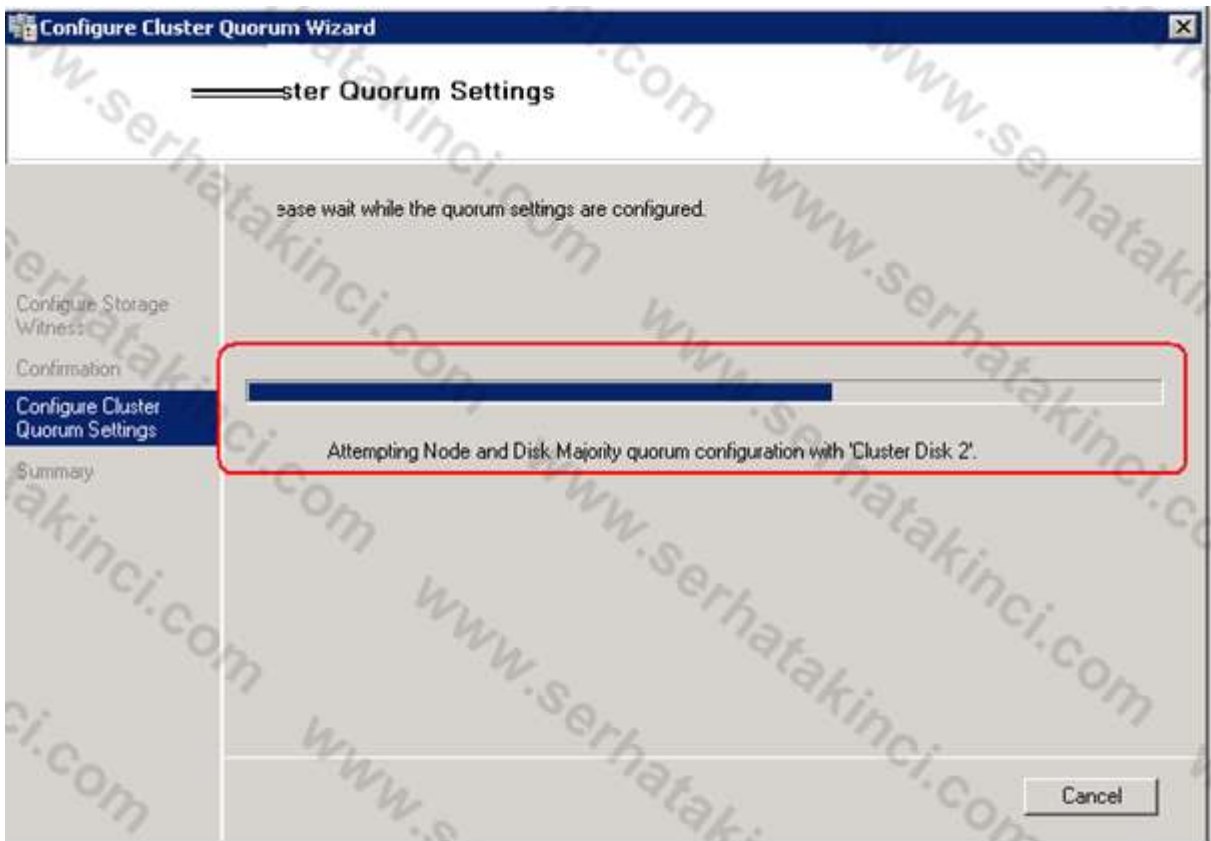
İlgili diski seçerek devam ediyoruz.



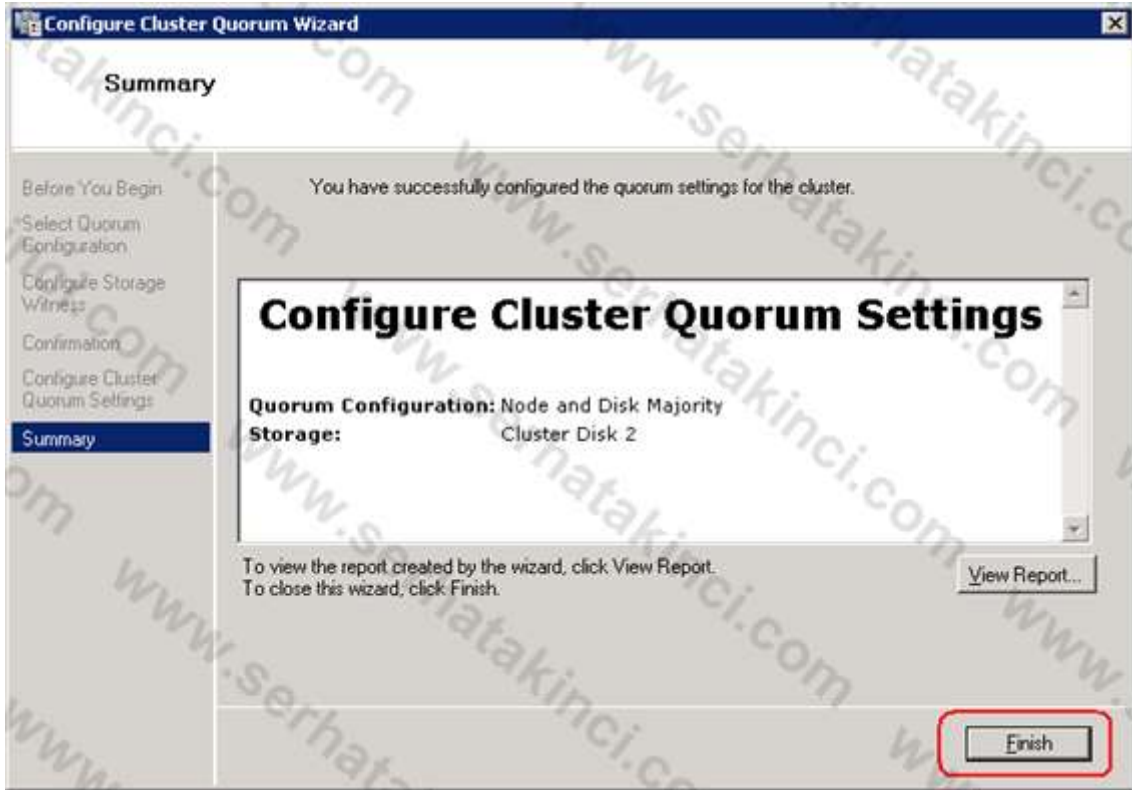
**Next** ile konfigürasyonu başlatıyoruz.



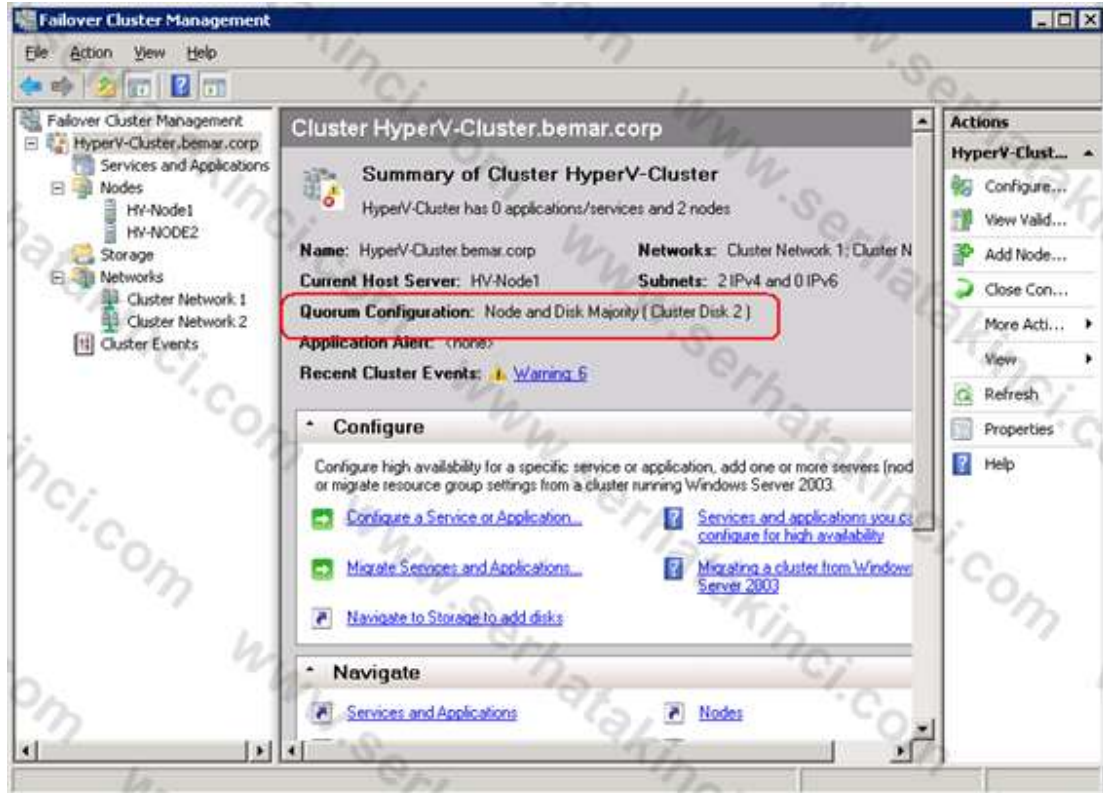
Quorum settings yapılıyor.



**Successfully** bilgisini görüyoruz ve quorum settings başarılı bir şekilde tamamlanmış oluyor.



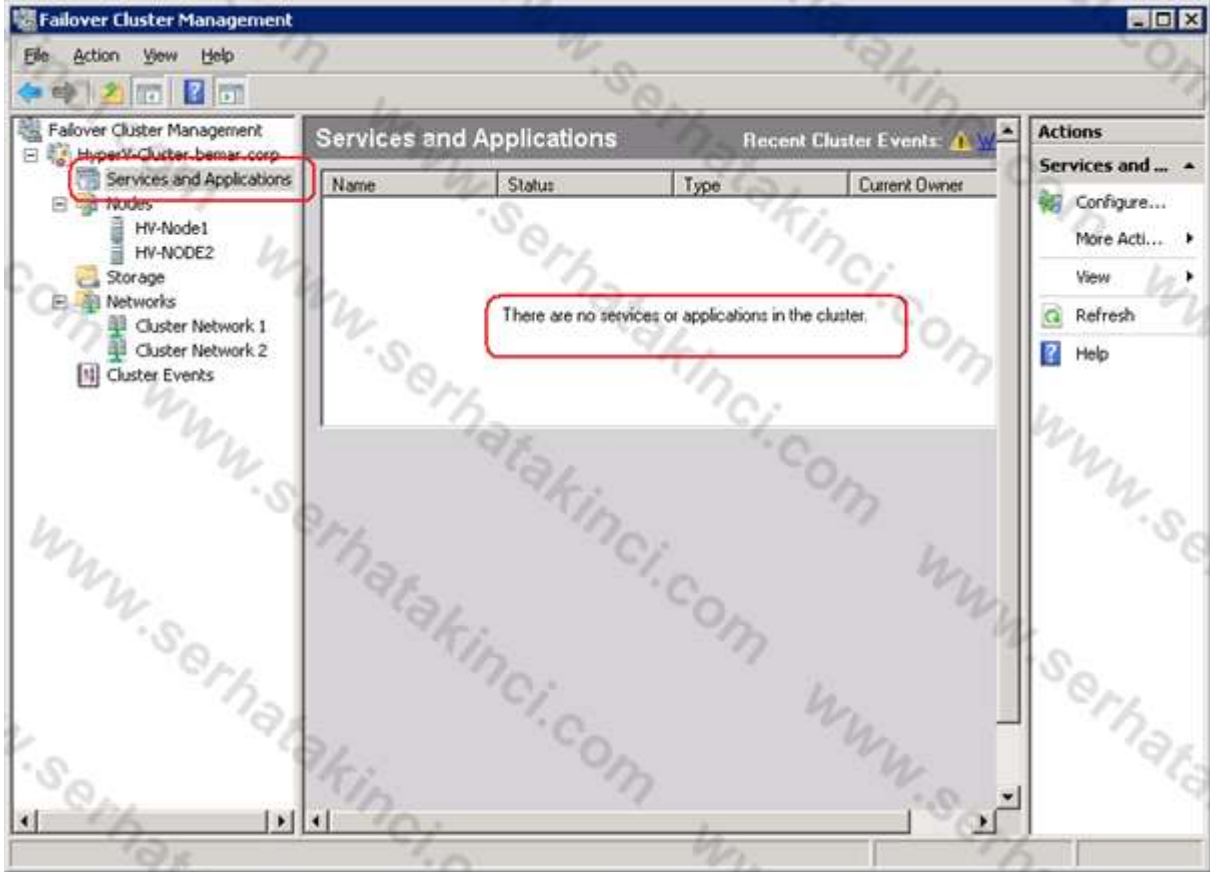
**Failover Cluster Management** konsoluna döndüğümüzde quorum tipini ve hangi disk üzerinde olduğu bilgisini **cluster summary** bölümünde görebiliyoruz.



## 9. Hyper-V Sanal Makinelerini Cluster Yapısına Dahil Etmek

Failover Cluster kurulumunu tamamlamış ve nodeları cluster yapısı içerisinde görüyor olmamız, sanal makinelerin cluster ortamında çalışmaya başlaması için yeterli değil.

Cluster kurulumu sonrasında ilgili servisin yada uygulamanın cluster yapısına ayrıca eklenmesi gerekiyor. Failover Cluster yönetim konsolunu açarsanız henüz herhangi bir uygulama/servis olmadığını görebilirsiniz.

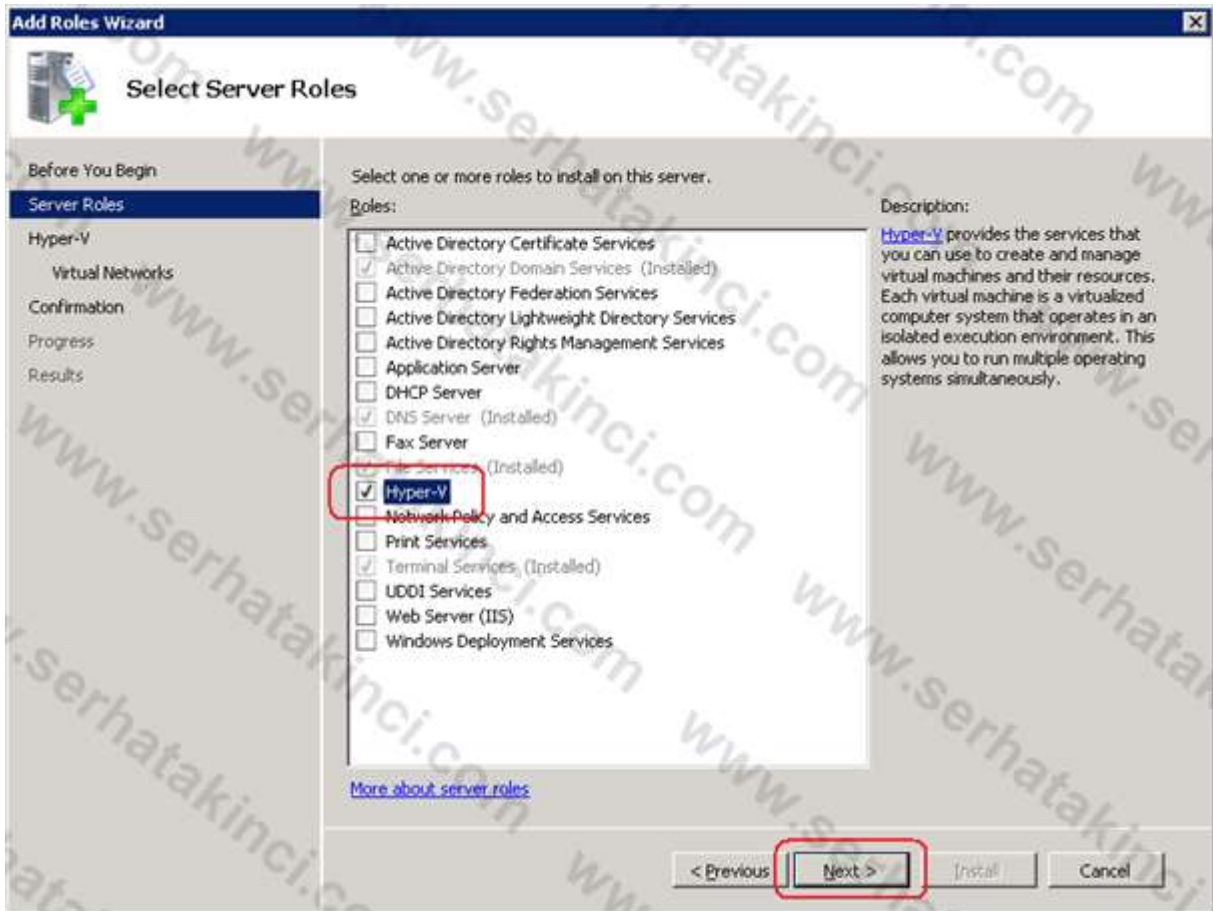


Yani cluster kurulumu yapmak, sanal makinelerin **High Available** olması için yeterli değil. Ayrıca henüz Nodelar üzerine Hyper-V kurulumu da yapmadık.

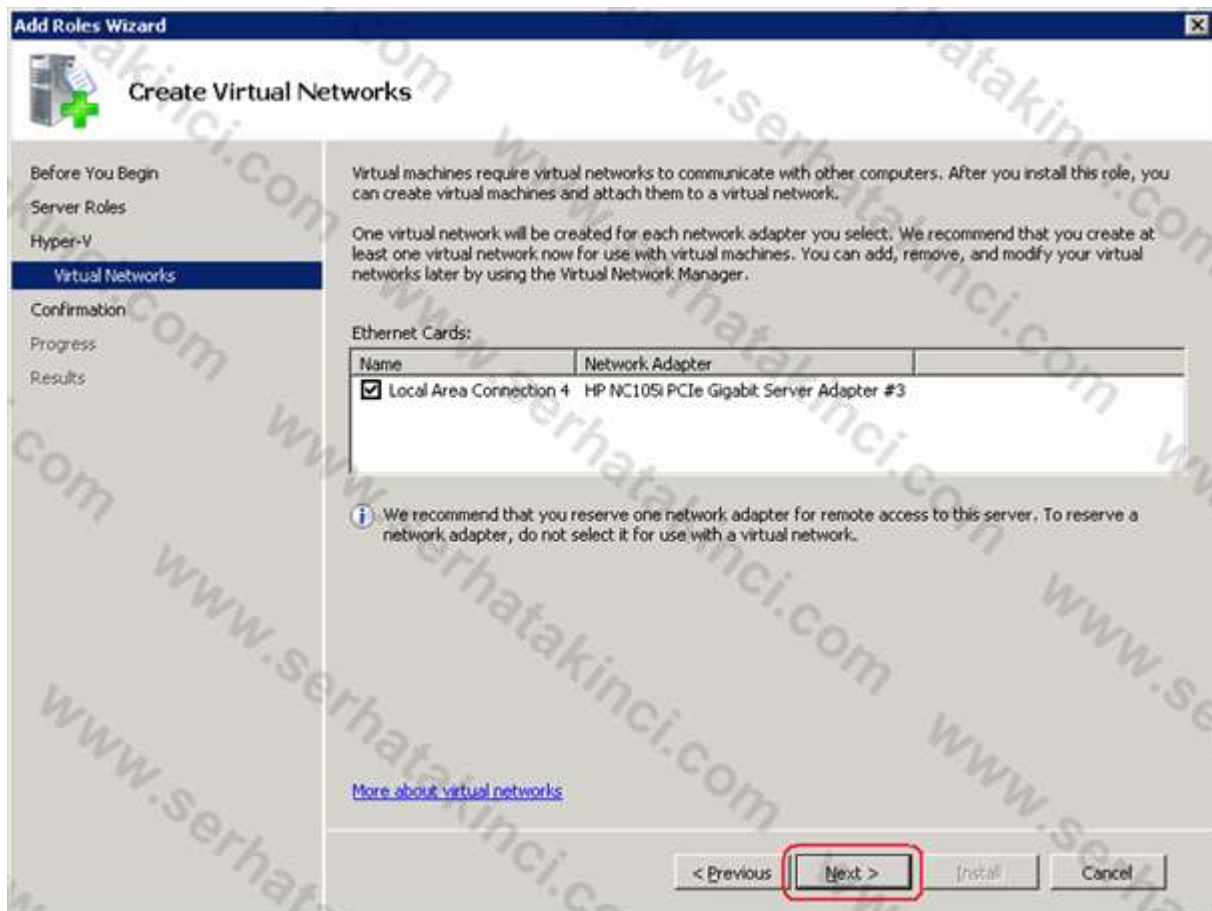
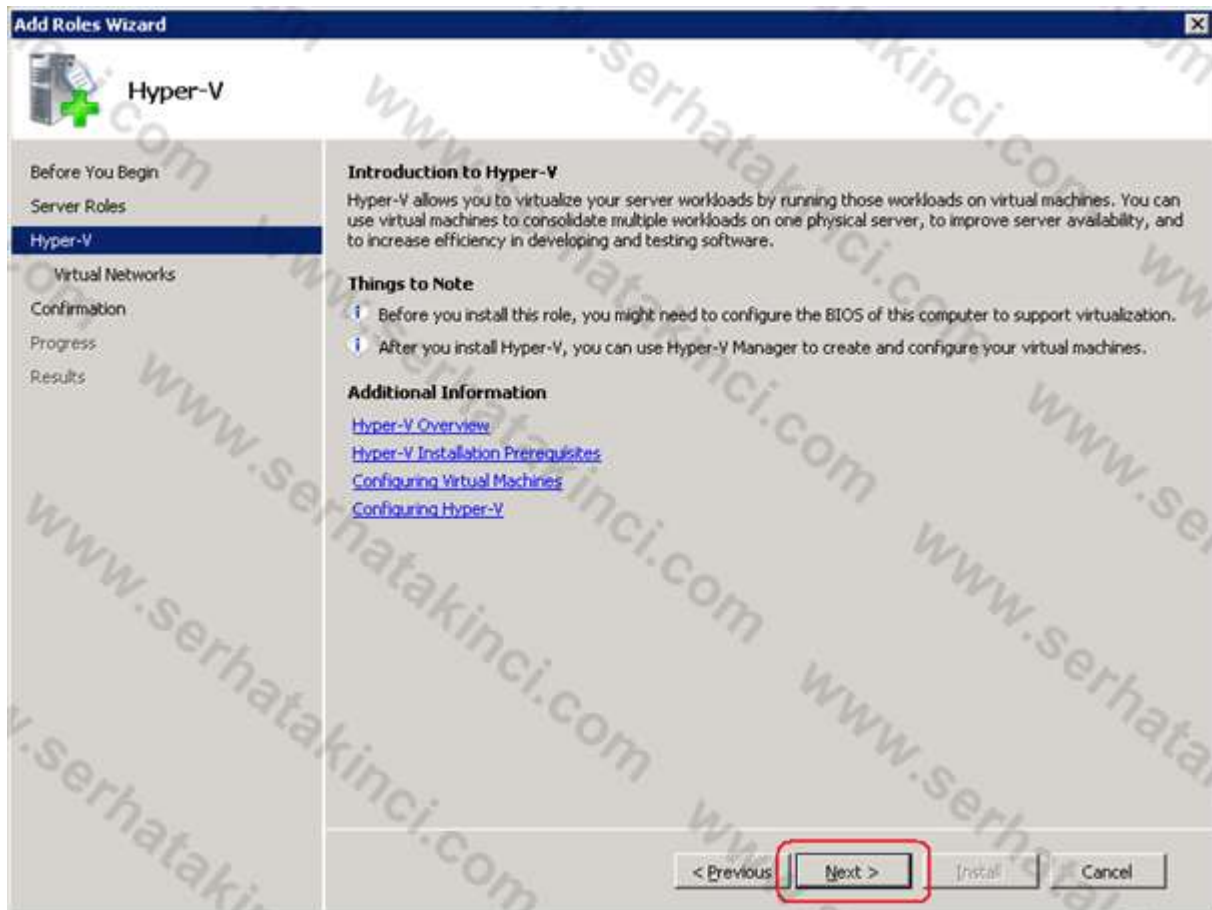
## 9.1. Nodelar Üzerine Hyper-V Kurulumu

Biz Hyper-V üzerinde çalışan sanal makineleri failover cluster yapısına almak istediğimiz için, servis ve uygulama olarak **Virtual Machine** ekleyeceğiz. Ama bunun öncesinde fiziksel Node'lar üzerine Hyper-V rolünü eklemeliyiz yani Hyper-V kurulumunu yapmalıyız. Hatırlarsanız Hyper-V kurulumunu cluster öncesinde veya cluster sonrası yapabileceğimizi söylemiştik.

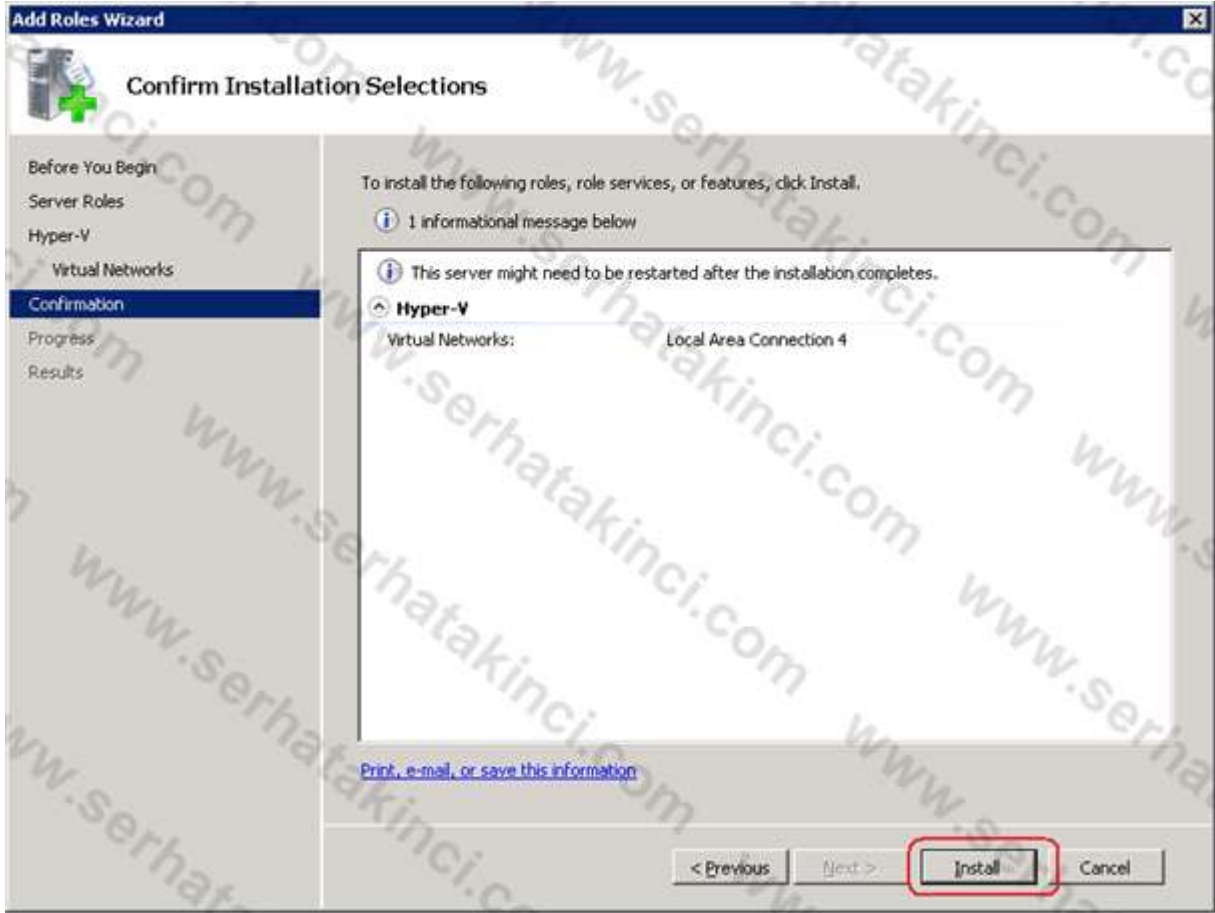
**Server Manager** konsolunda, **Roles** bölümünde, **Hyper-V** rolünü seçerek ileri dedikten sonra wizard yardımı ile birkaç ayar yaparak Hyper-V rolünü eklemeniz mümkün.











Hyper-V kurulumu ile ilgili fazla detaya girmiyorum. Bu konunun ayrıntıları için daha önceki makalelerime bakabilirsiniz.

Her iki fiziksel node üzerine (HV-Node1 ve HV-Node2) Hyper-V kurulumunu doğru olarak yaptıktan ve çalıştırdıktan emin olduktan sonra devam edebiliriz.

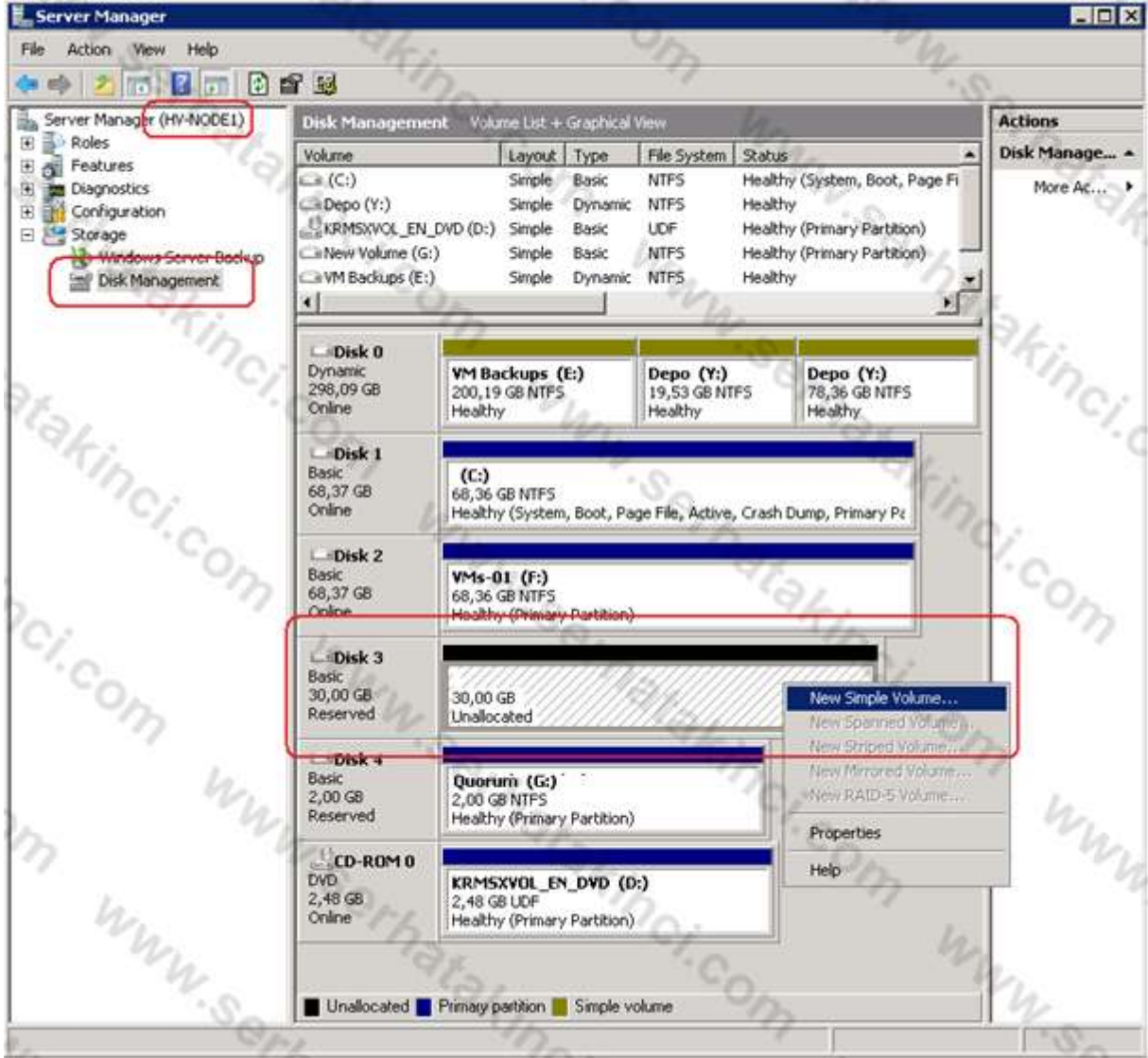
## 9.2. Test için Sanal Makine (VM) Yaratılması

Test amaçlı yeni bir VM (sanal makine) yaratalım ve nodeların bu VM'i çalıştırıp çalıştıramadığını kontrol edelim.

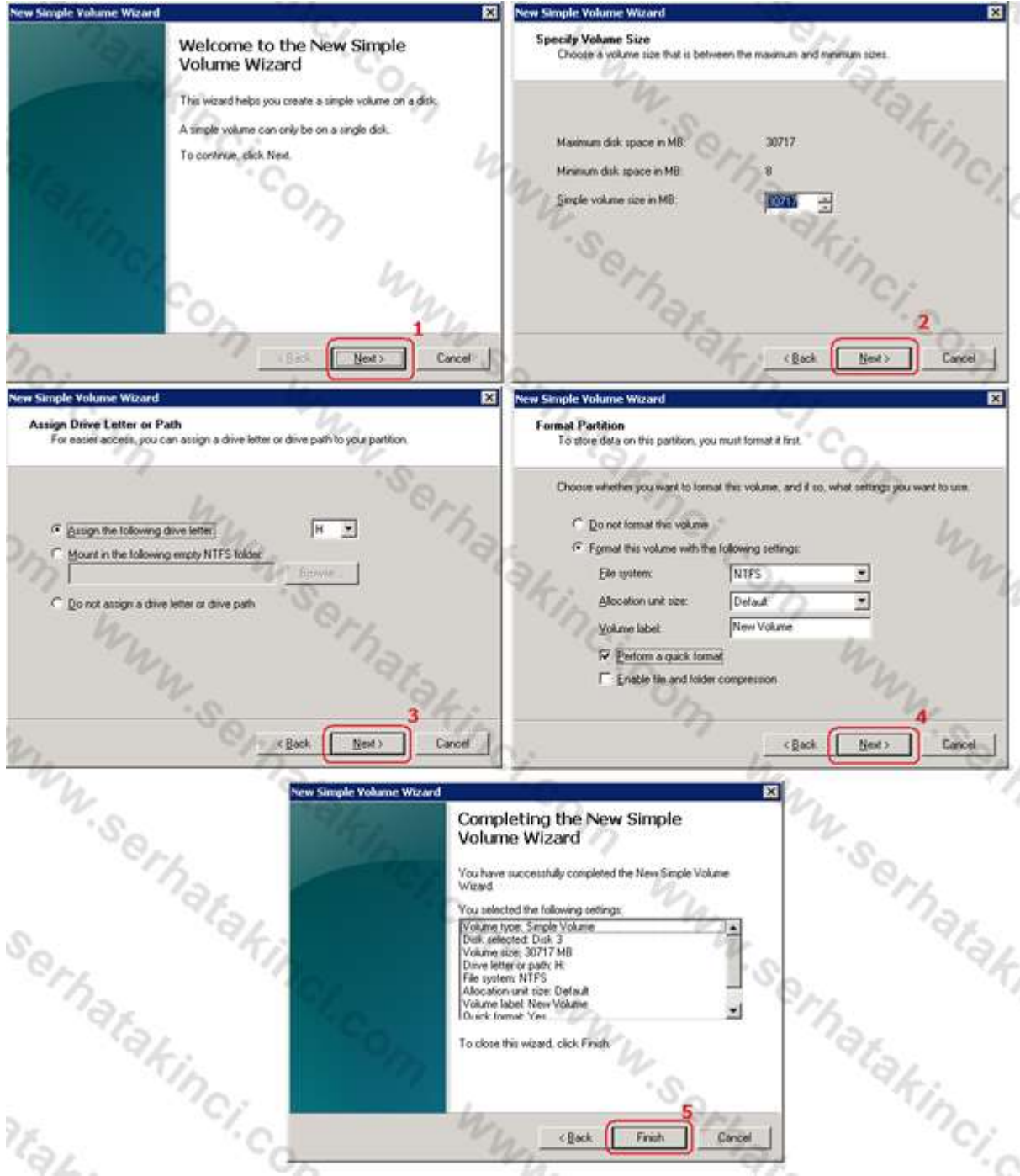
Failover Cluster yapısına almak istediğimiz VM'lerin storage üzerindeki ortak birimlerde duruyor olması ve cluster nodelarının bu birime/birimlere erişebiliyor olması gerektiğinden daha önce bahsetmiştik.

Hatırlarsanız storage üzerinde iki Virtual Disk yaratıp bunları node'lar üzerine atamıştık. Birini cluster bilgisini tutan **Quorum** için kullandık (2GB olan). Diğer disk ise hala boşta (30GB olan). Boşta olan disk VM için kullanacağız.

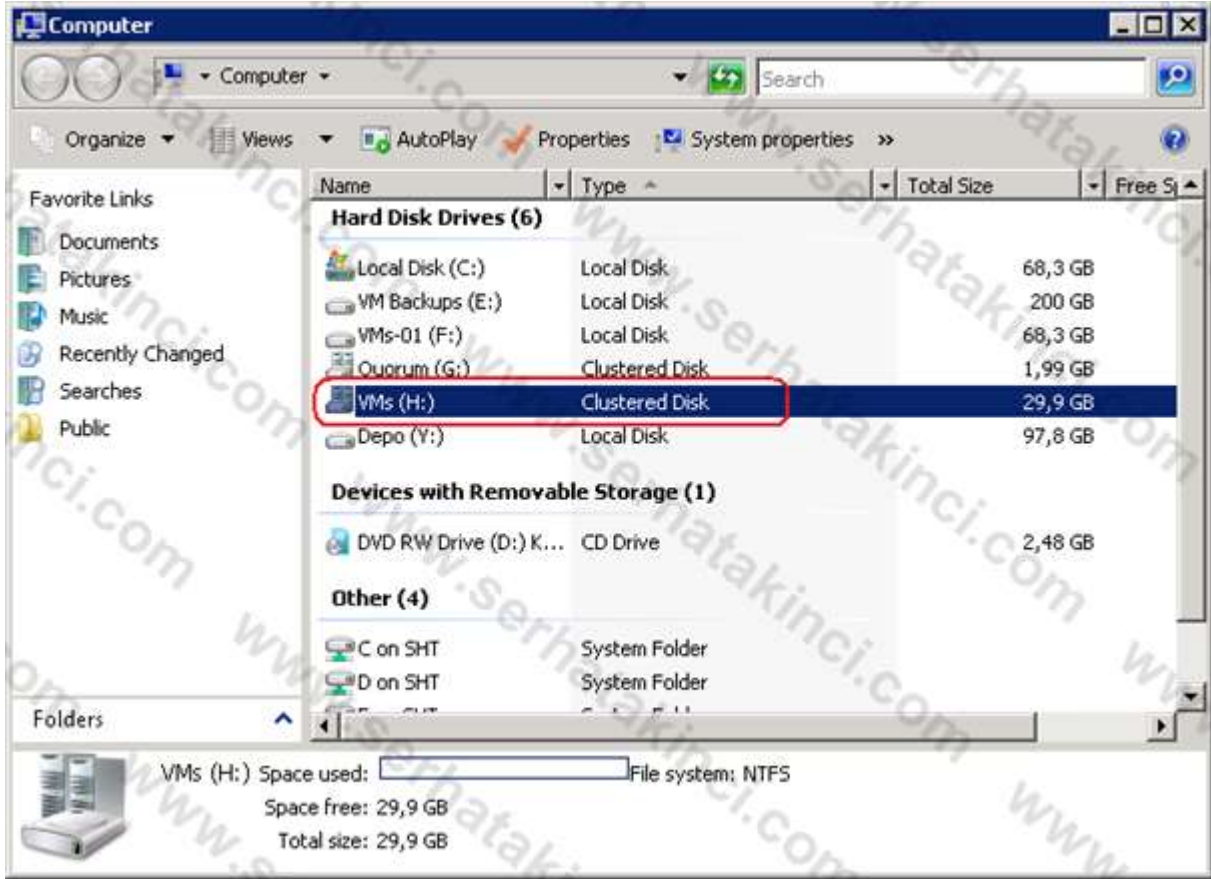
Öncelikle **HV-Node1** üzerinde **Disk Management**'i açıyoruz ve bu disk üzerinde NTFS formatlı bir partition oluşturuyoruz.



Wizard yardımı ile partition oluşturma işlemini tamamlayalım.



**HV-Node1** üzerinde **Computer** penceresinde oluşan volume'u görebiliriz. Ben ismini **VMs** olarak değiştirdim.

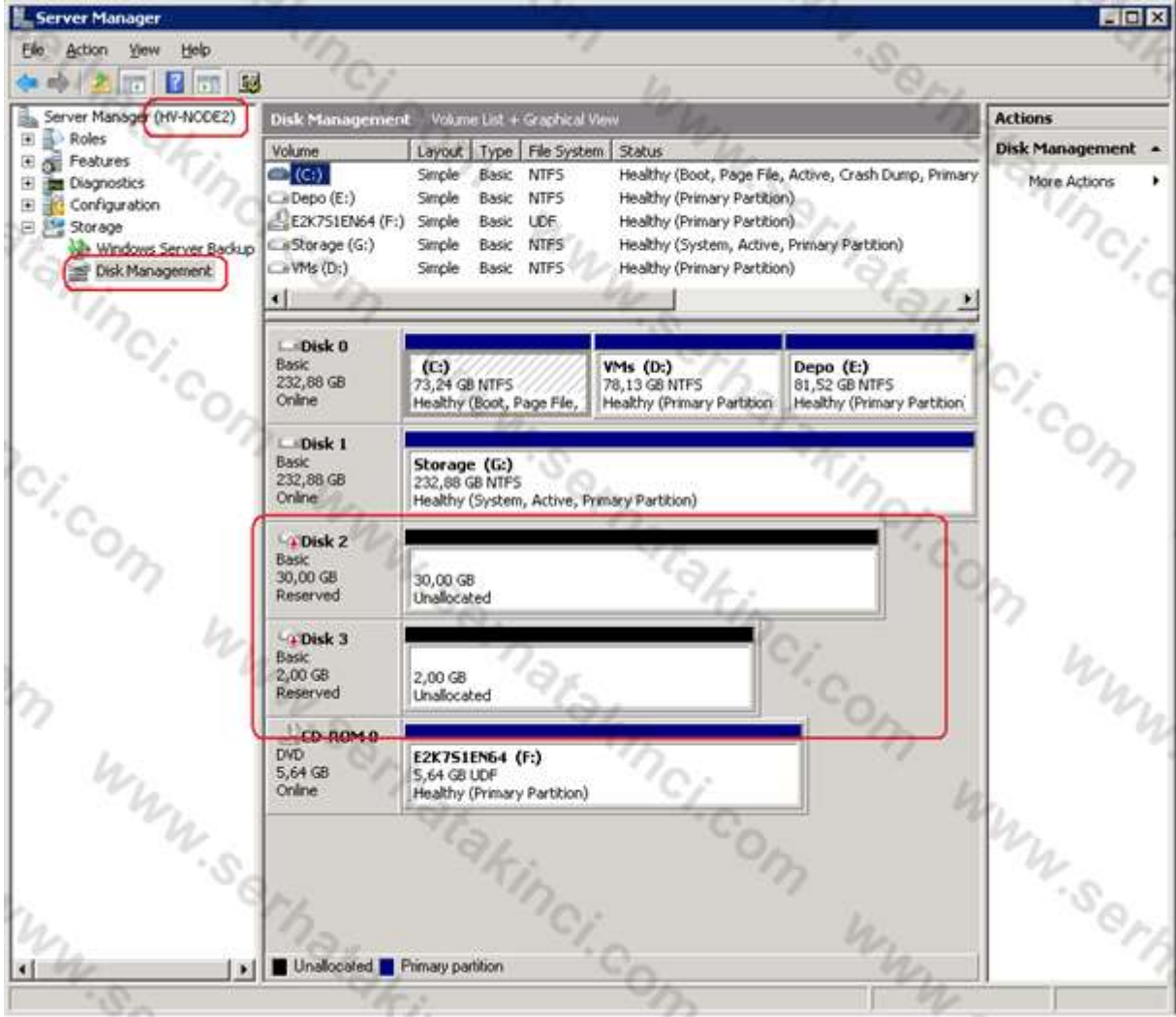


Dikkat ederseniz **Type** olarak **Clustered Disk** görünüyor.

Hemen üstünde ise daha önce biçimlendirdiğimiz ve quorum bilgisinin tutan **Quorum** bölümünü de görebiliyoruz.

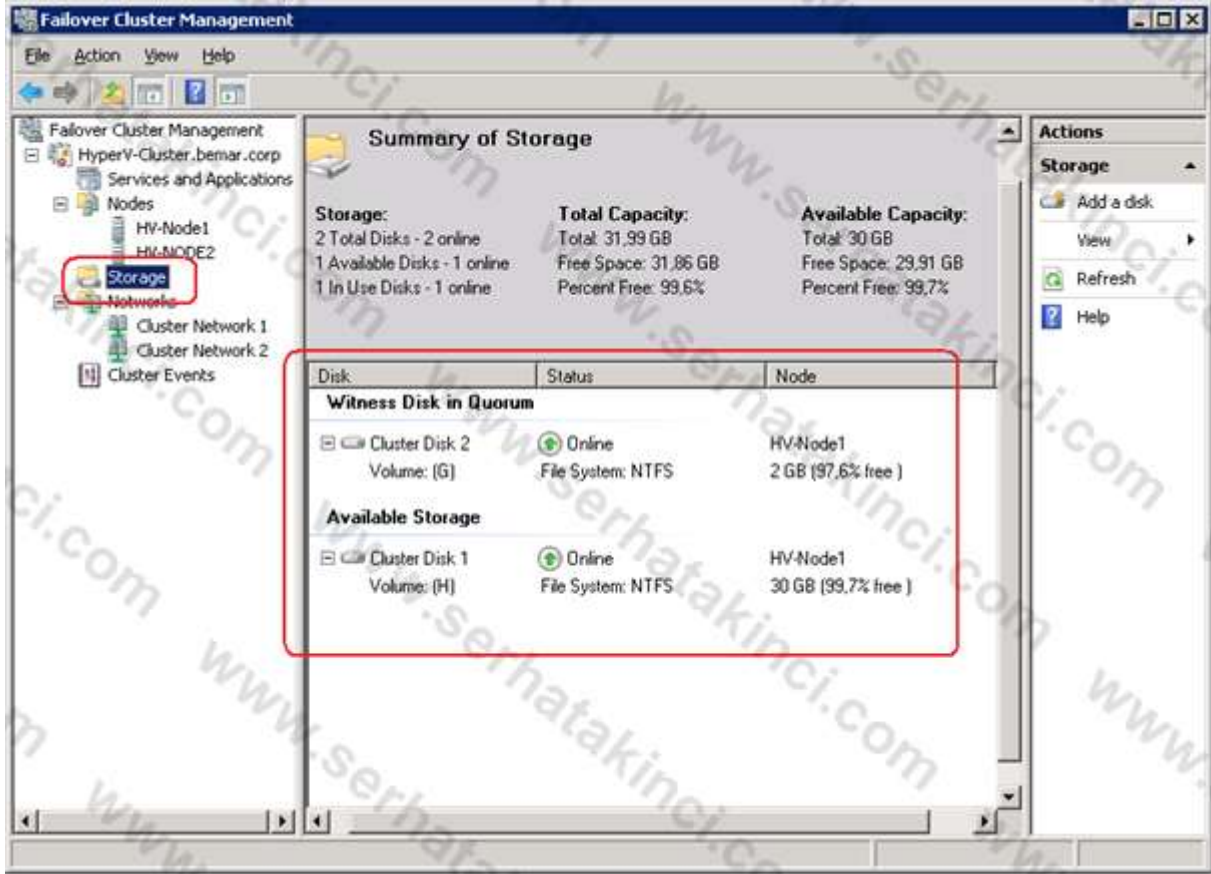
Peki şu an **HV-Node2** üzerinde durum ne?

**HV-Node2** üzerinde **Server Manager > Disk Management** bölümüne gelerseniz, HV-Node1 üzerinde biçimlendirdiğimiz disklerin HV-Node2 üzerinde **Reserved** durumda olduğunu görebilirsiniz.



Yine HV-Node1'e dönelim ve **Cluster Management** konsolundaki duruma bakalım.



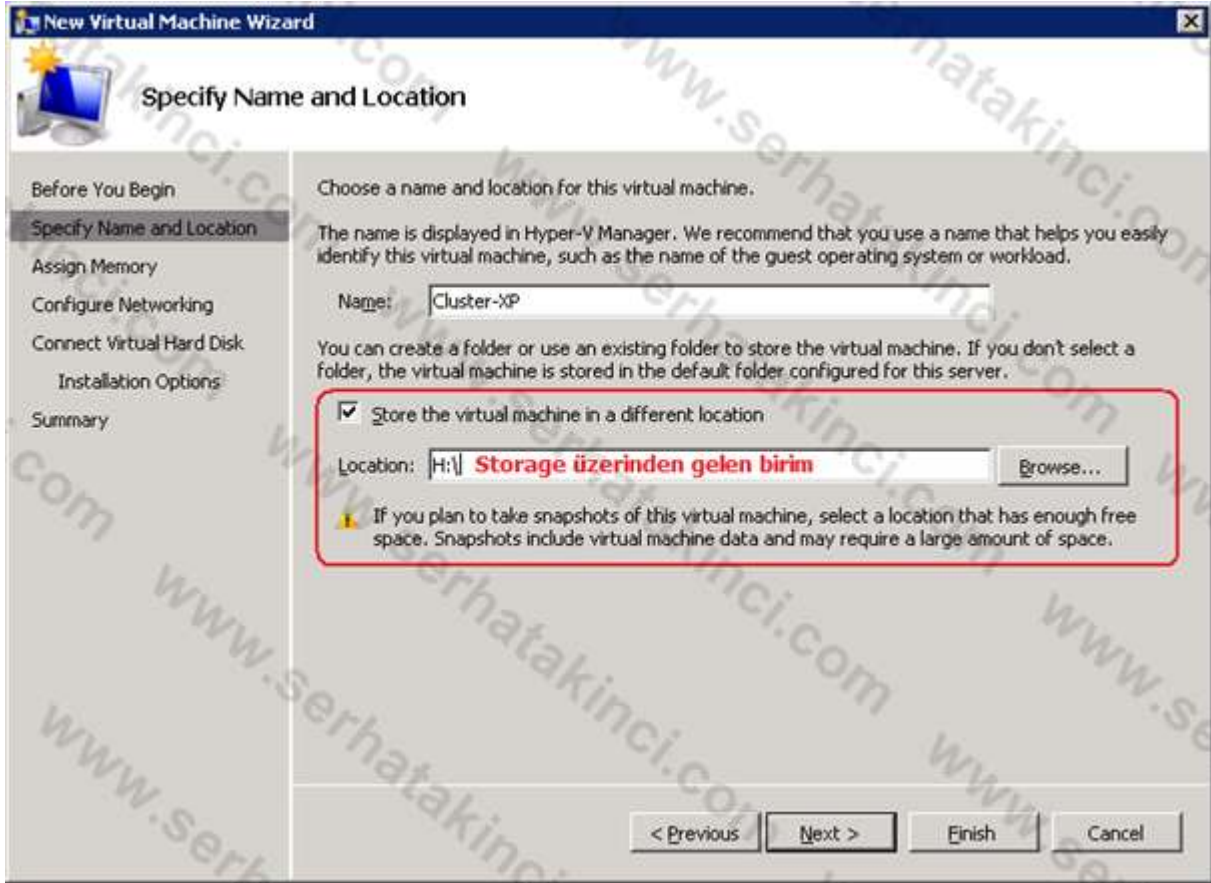


Gördüğünüz gibi iki diskte **storage** bölümünde listeleniyor. Disklerin şu anki sahibi ise **HV-Node1**.

Şimdi HV-Node1 üzerinde yeni bir VM yaratalım ve bu VM storage üzerinden atadığımız diskte bulunan **VMs (H:\)** partition'ı içerisinde dursun.

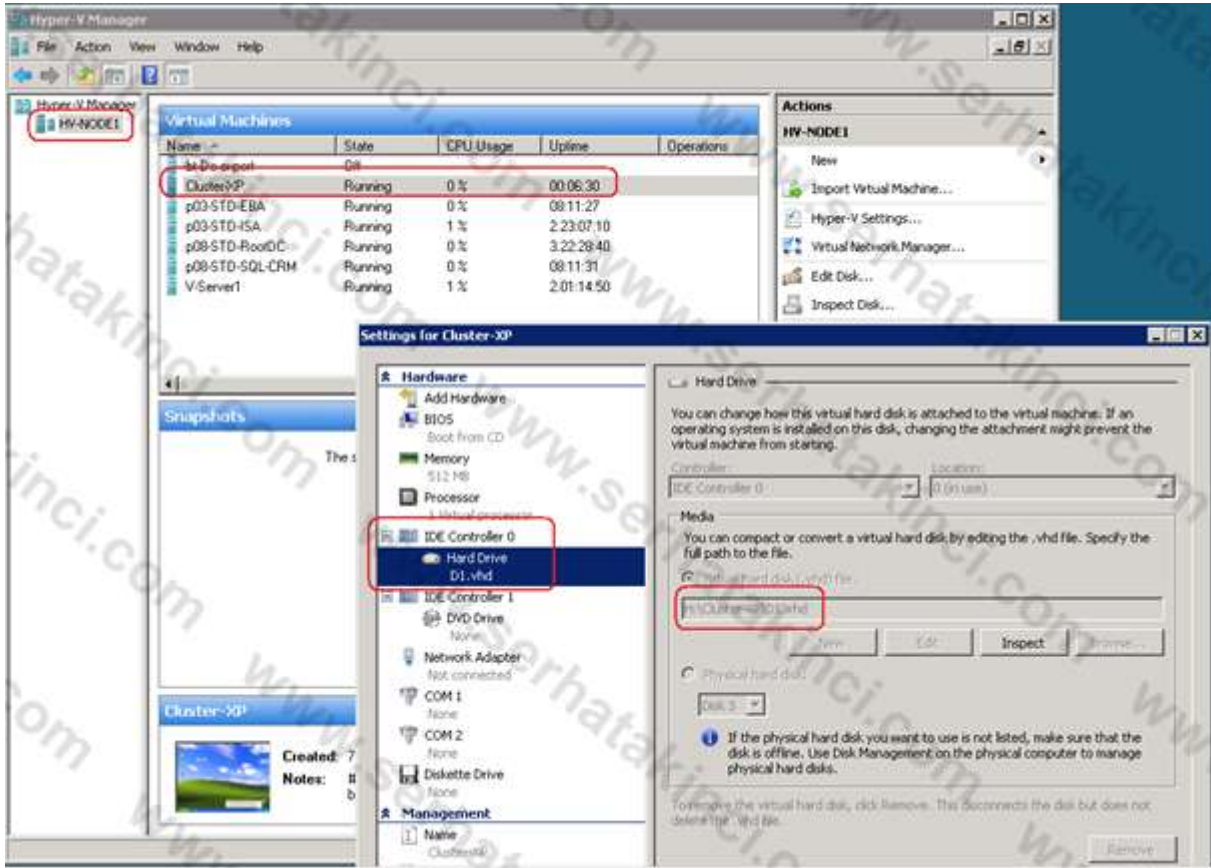
Sıradan bir VM yaratmaktan farkı yok. Sadece VM'in duracağı yer olarak storage üzerinden gelen **clustered disk**'i göstermemiz gerekiyor.





VM yaratmak ile ilgili ayrıntıya girmiyorum. Eğer nasıl yaratılacağını bilmiyorsanız, bu konu için daha önce yazmış olduğum makalelere göz atabilirsiniz.

VM'i yarattıktan ve içerisine **Guest OS** kurduktan sonra aşağıdaki gibi görünüyormalı.



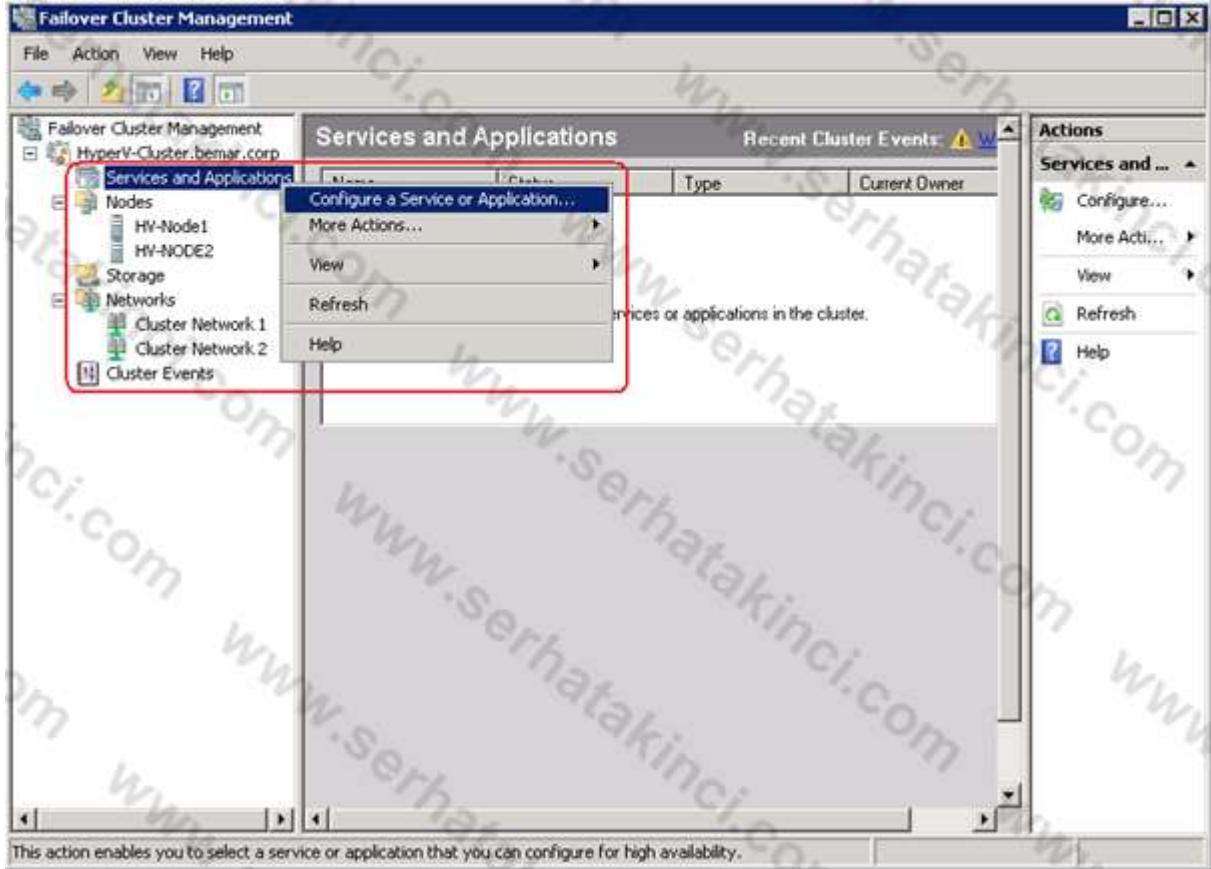
Yukarıda gördüğünüz gibi **Cluster-XP** isimli sanal makine, HV-Node1 üzerinde çalışıyor. Ama sanal makinenin **vhd** formatlı sanal diski ve diğer ilgili dosyaları **H:\** yani **clustered disk** üzerinde duruyor.

Şimdi Failover Cluster yapımıza **Services and Applications** olarak **Virtual Machine** ekleyeceğiz ve yarattığımız VM'i **High Available** yapacağız.

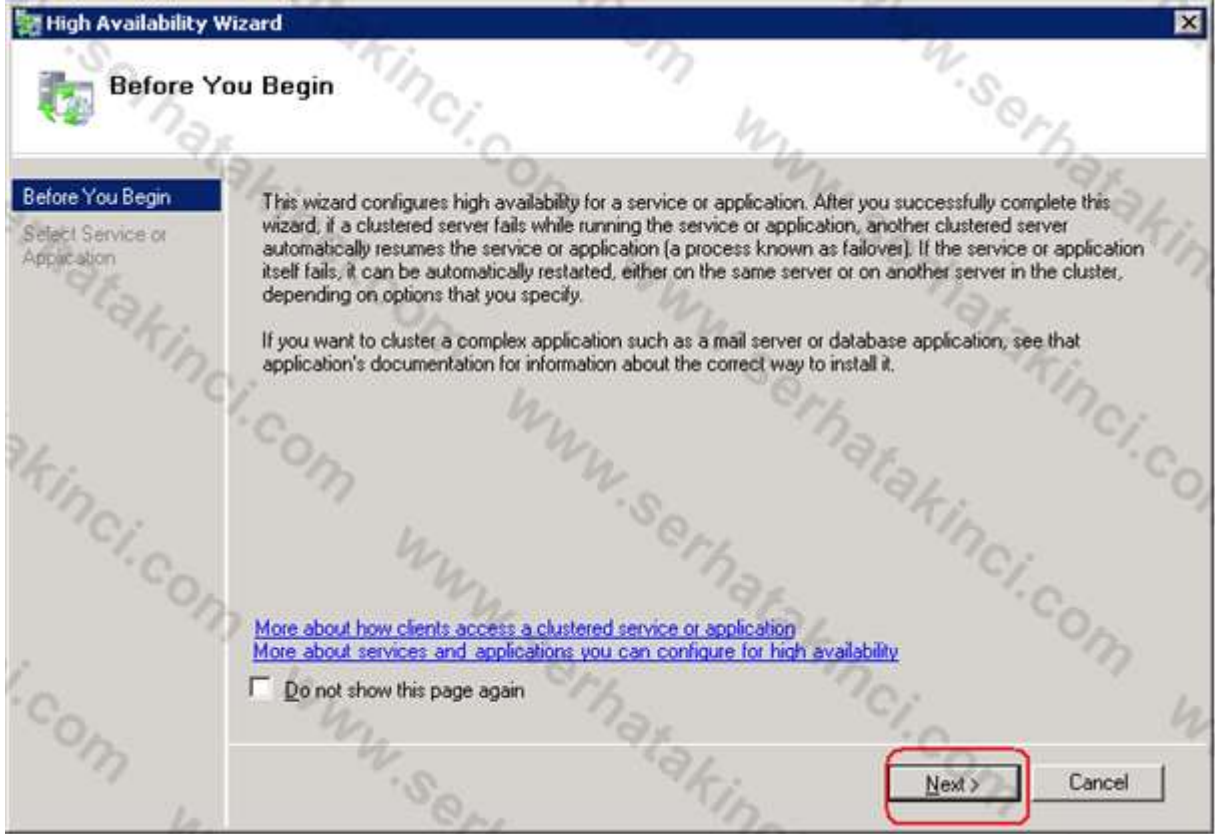
### 9.3. Failover Cluster Yapısına Virtual Machine Servisini Ekleme

HV-Node1 üzerinden **Failover Cluster Management** konsolunda **Services and Applications** üzerinde sağ tıklıyoruz ve **Configure a Service or Application** diyoruz.

**High Availability Wizard** açılıyor.

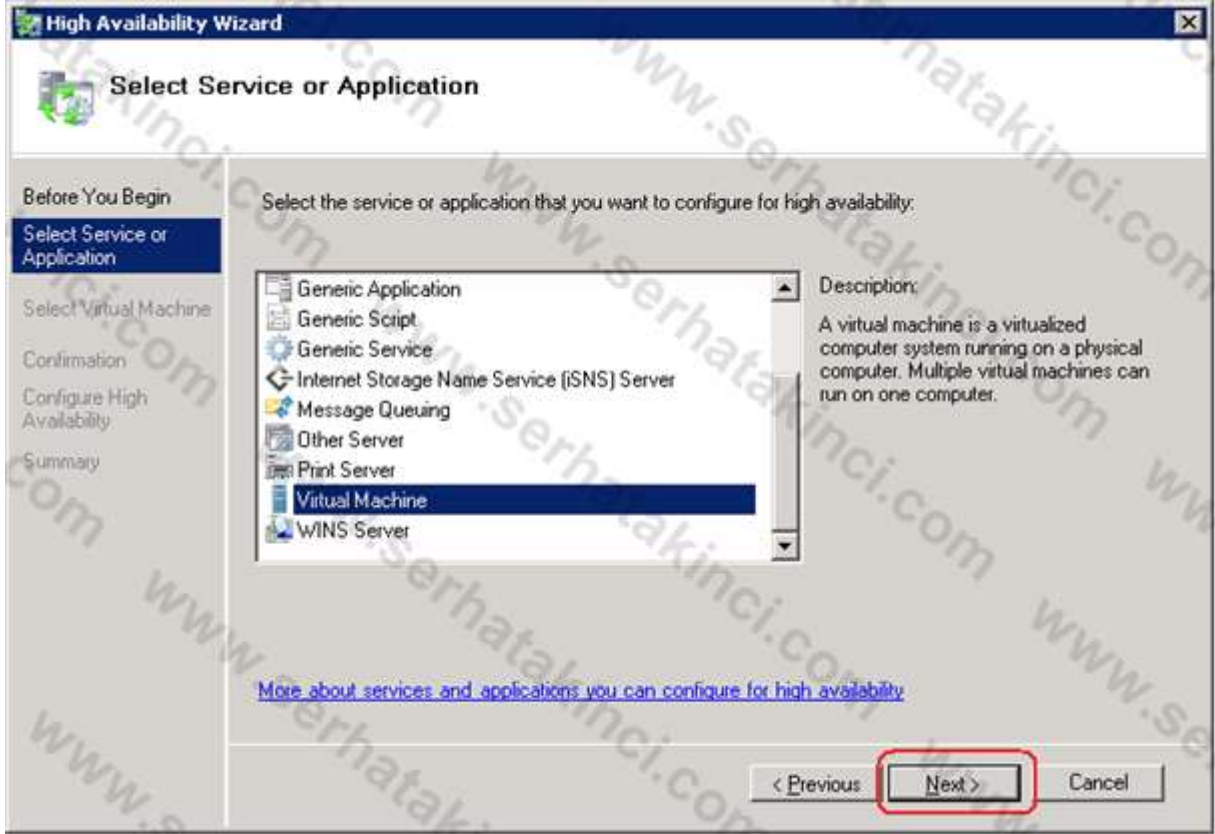


**Next** diyoruz.



**Select Service or Application** penceresinde gördüğünüz gibi çok sayıda servis ve uygulama var.

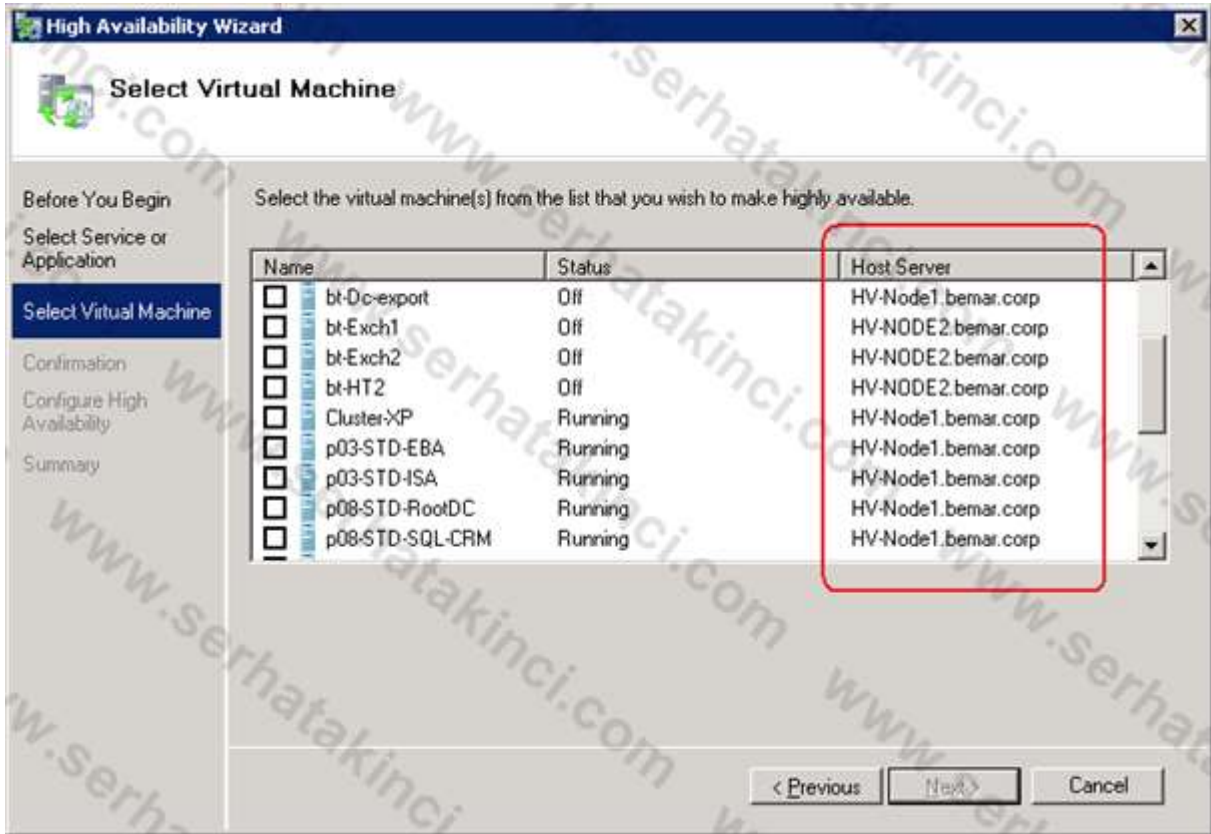
Biz **Virtual Machine** seçerek devam ediyoruz. Bu noktada listedeki farklı uygulamalar için de cluster yapıları oluşturmanız mümkün.



**Next** dedikten sonra gelen **Select Virtual Machine** penceresinde ise cluster yapısına dahil edeceğimiz VM'i/VM'leri seçiyoruz. Yani Node'lar üzerindeki tüm VM'ler hemen cluster yapısına dahil olmuyor. Hangi VM'in cluster içerisinde çalışacağına bu ekranda seçim yaparak biz karar veriyoruz. Tabii ki tümünü seçme şansımız da var.

Dikkatinizi çekmek istediğim bir diğer nokta ise bu pencerede cluster node'ları üzerindeki tüm VM'lerin listeleniyor olması (aşağıda da görünüyor). Yani HV-Node1 ve HV-Node2 üzerindeki tüm VM'ler bu listede yer alıyor ve hepsini seçebiliyoruz. Ama seçtiğimiz VM'in clustered disk üzerinde olması gerektiğini unutmayın. Yani burada listeleniyor olması, cluster yapısına dahil edebileceğimiz anlamına gelmiyor. Clustered disk şart.

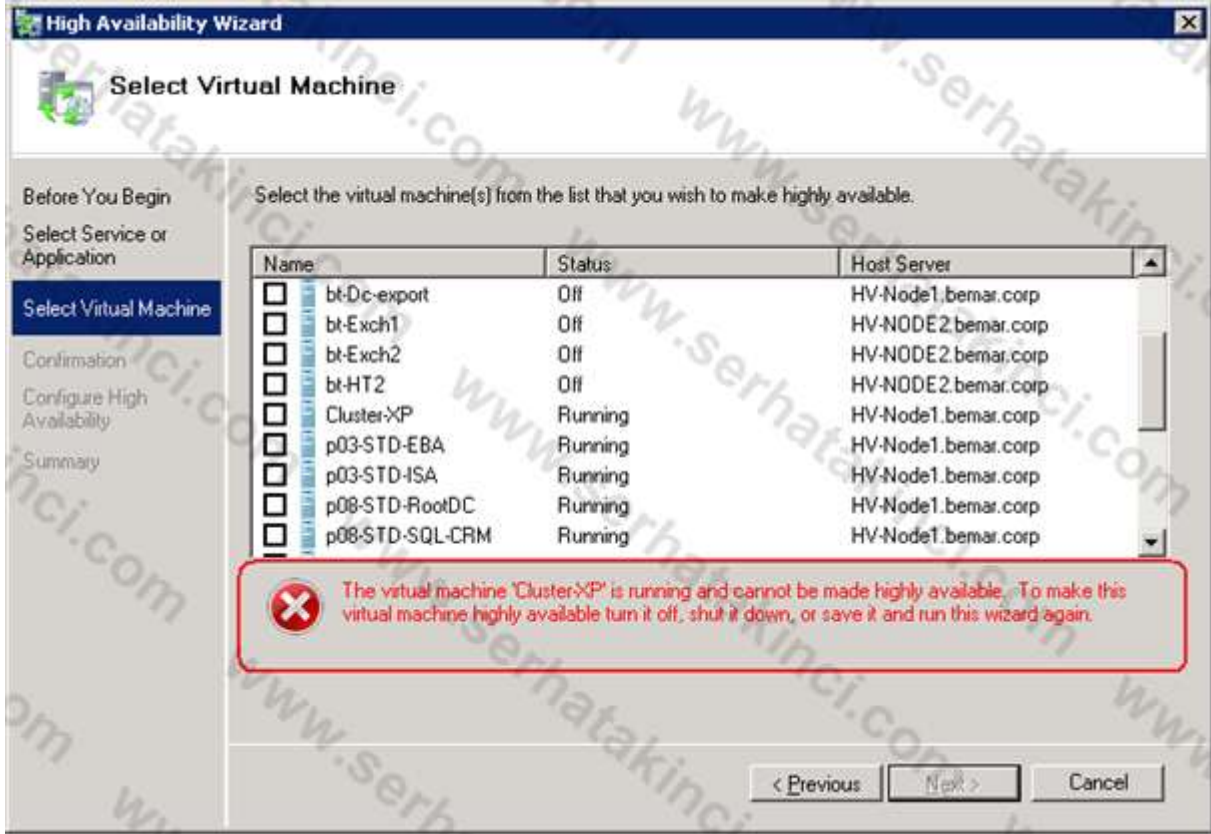




Bizim clustered disk üzerinde duran VM'in ismi **Cluster-XP**. Bu nedenle onu seçerek devam ediyorum.

Ama **Cluster-XP** seçtiğimde bir hata alıyorum.

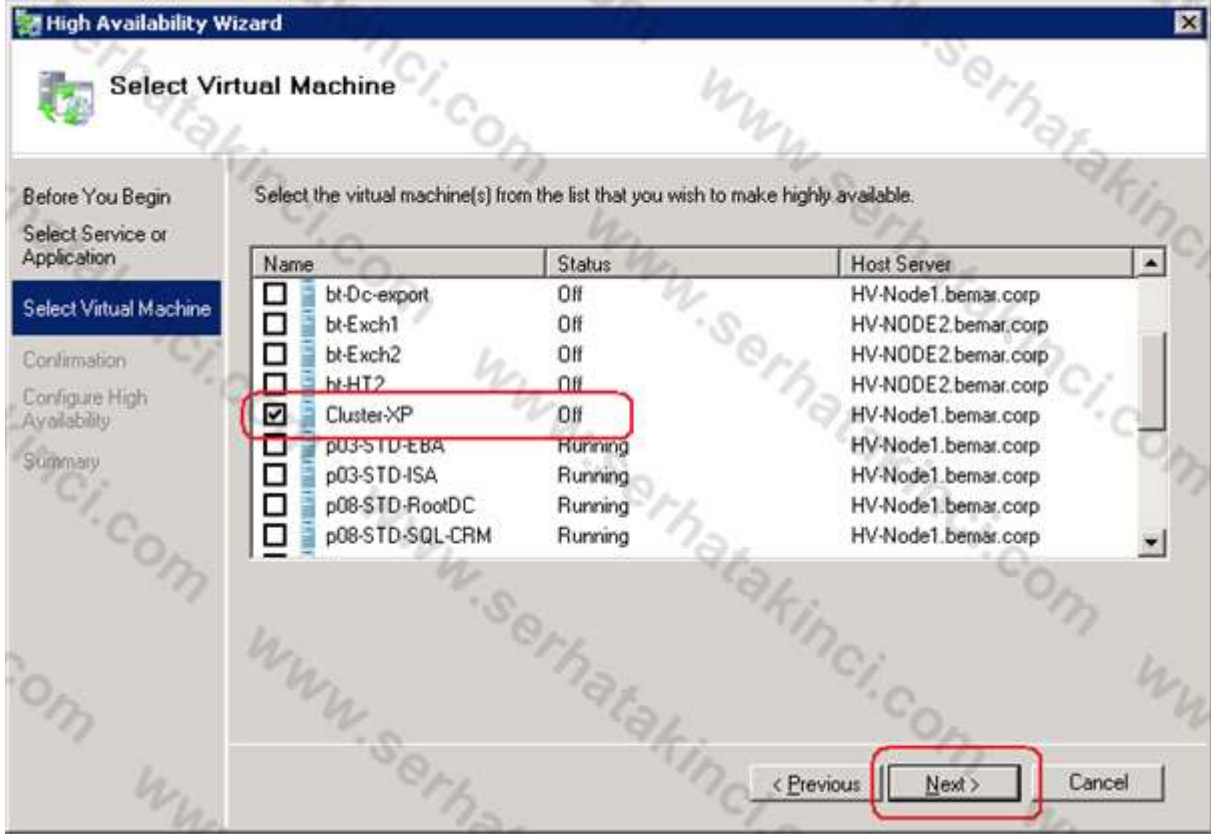




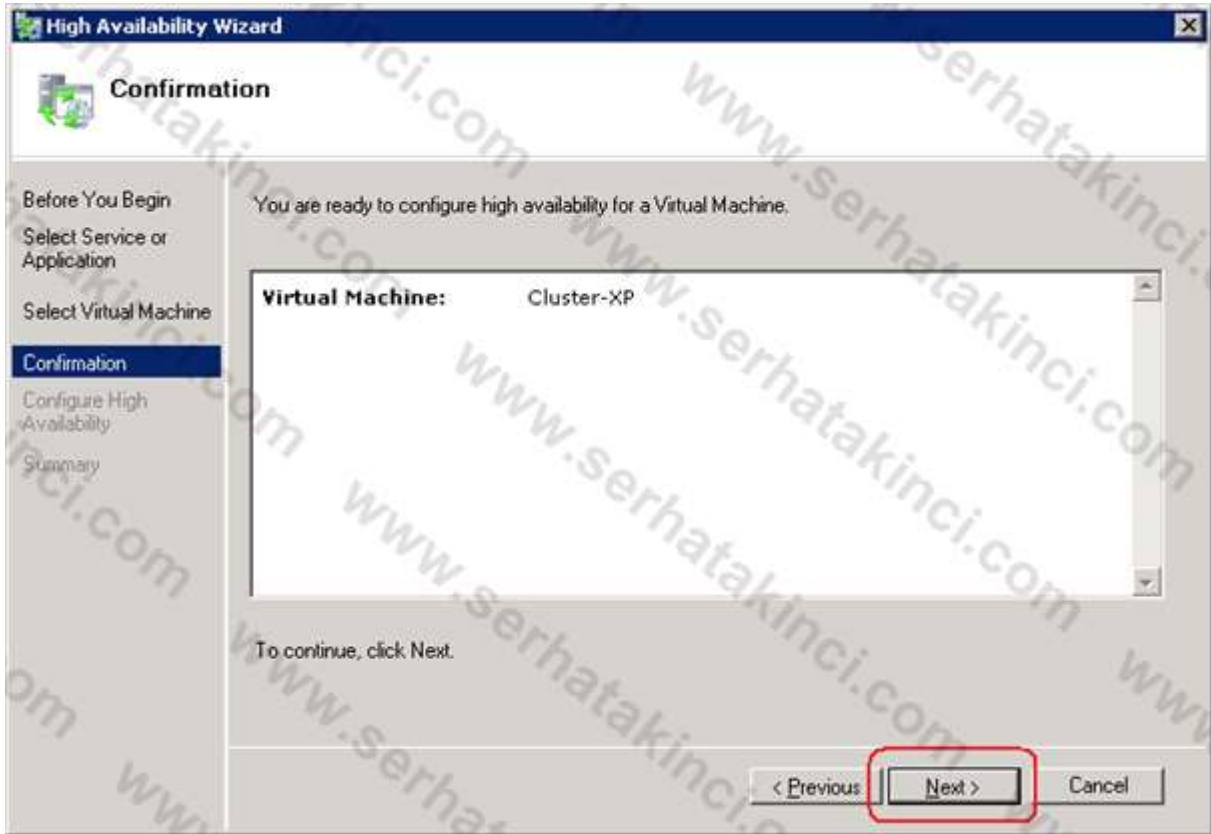
Bunun anlamı şu: bu konfigürasyonu, cluster yapısına dahil edeceğimiz VM çalışır durumdayken yapamıyoruz. VM'in **off** durumda olması gerekiyor.

VM'i **Hyper-V Manager** konsolundan ya da ilgili ara yüz üzerinden **off** yaptıktan sonra tekrar **High Availability Wizard**'ı açıyoruz ve aynı yere kadar ilerliyoruz.

**Cluster-XP**'i seçerek **Next** diyoruz.



**Next** ile konfigürasyonu başlatıyoruz.



İşlem sürüyor.



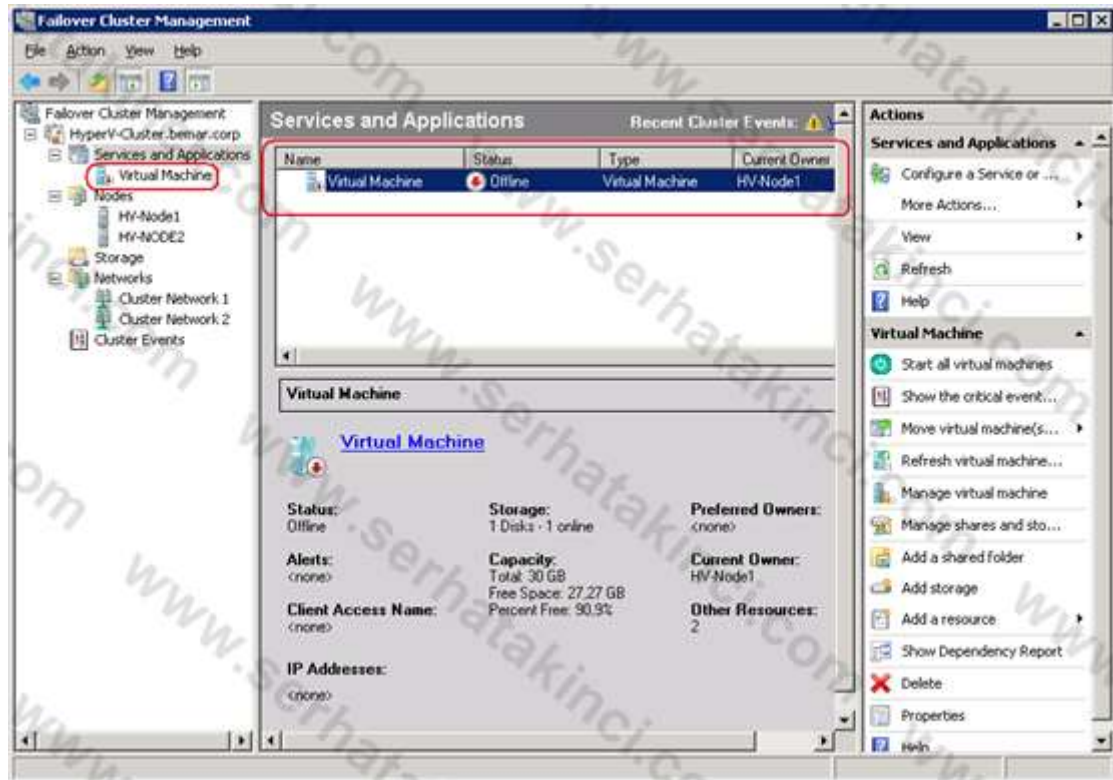
**Summary** penceresinde **Success** mesajını gördükten sonra yapılandırma başarılı bir şekilde tamamlanmış oluyor.

**Finish** diyerek bitiriyoruz.



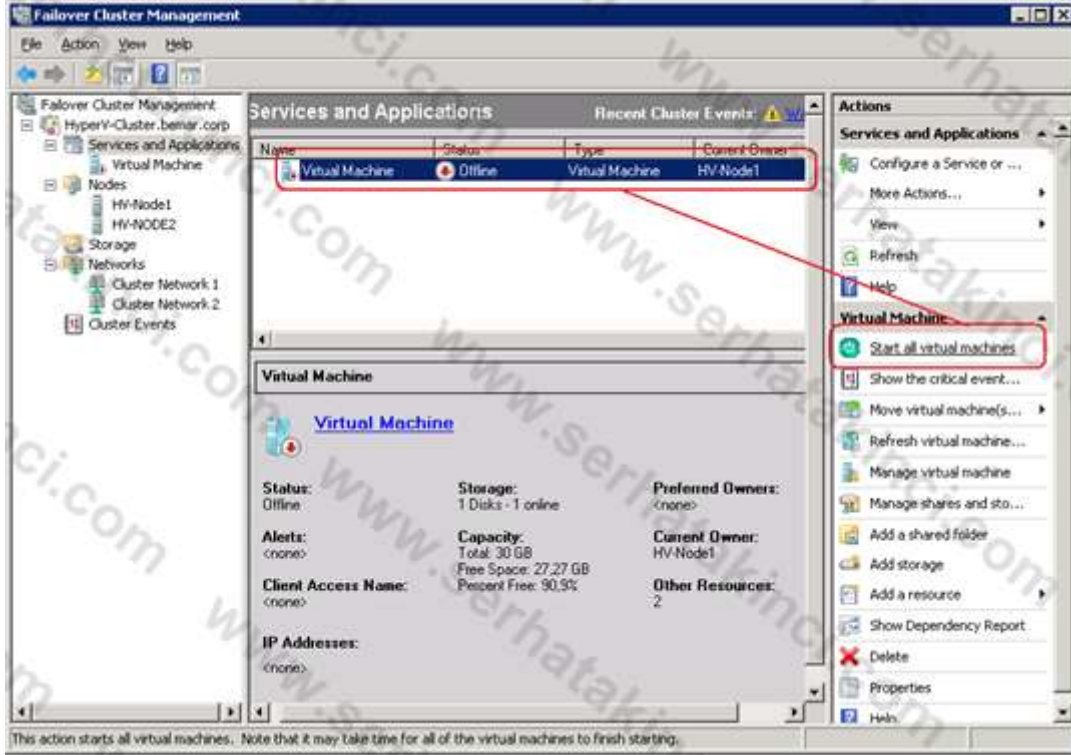
Artık **Failover Cluster Management** konsoluna döndüğümüz zaman **Services and Applications** altında **Virtual Machine** servisini görebiliriz ve şu an offline durumda.

Bu servis için **Current Owner** olarak ise **HV-Node1** görünüyor.



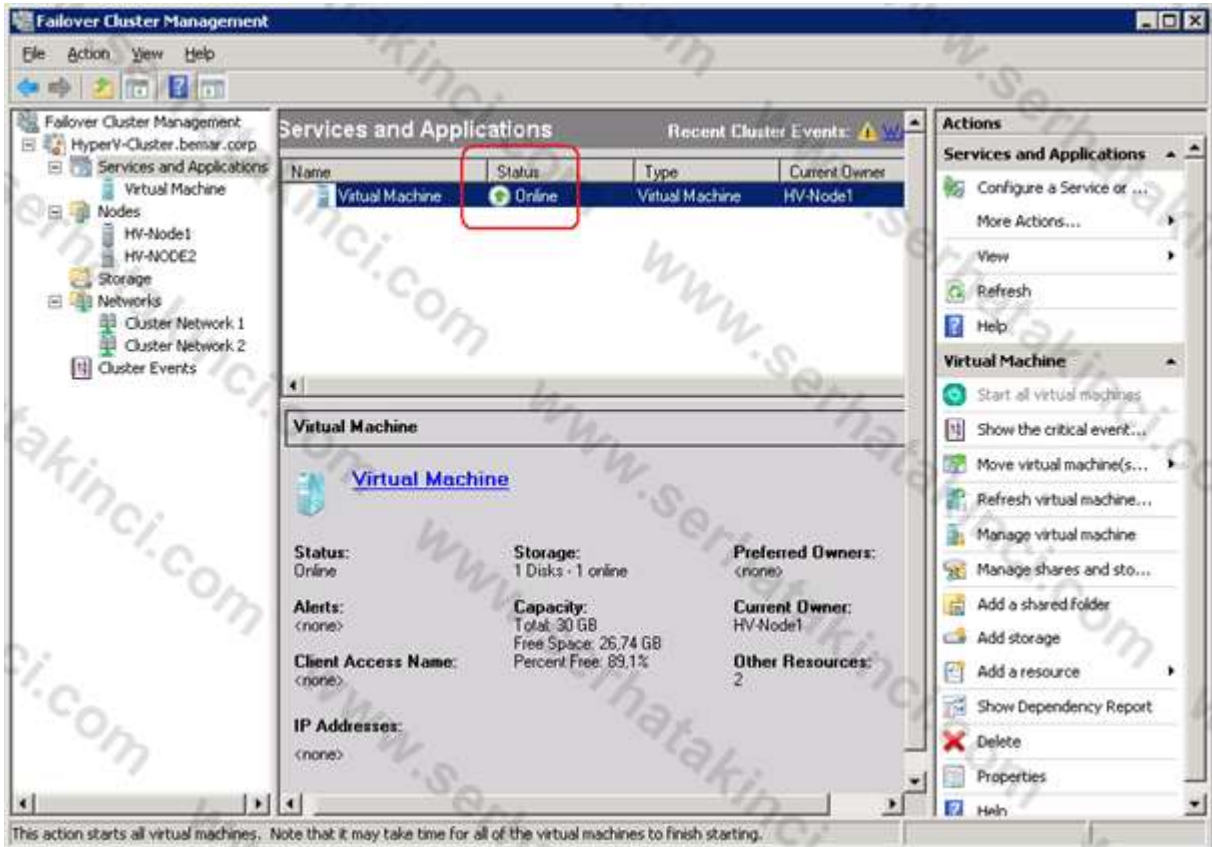
Hemen start edelim.





Bu komut ile Virtual Machine servisi içerisinde ekli olan tüm VM'leri start etmiş oluruz.

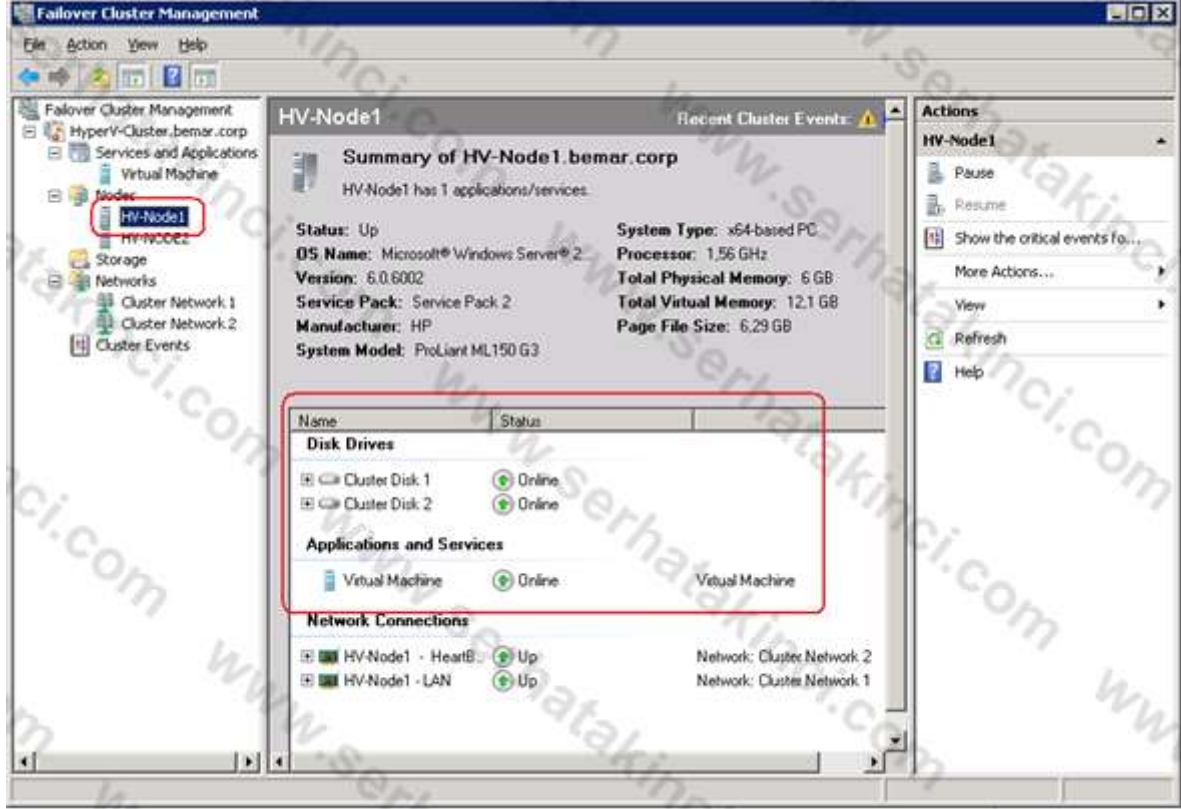
Biz tek bir VM eklediğimiz için şu an tek bir VM start oldu.



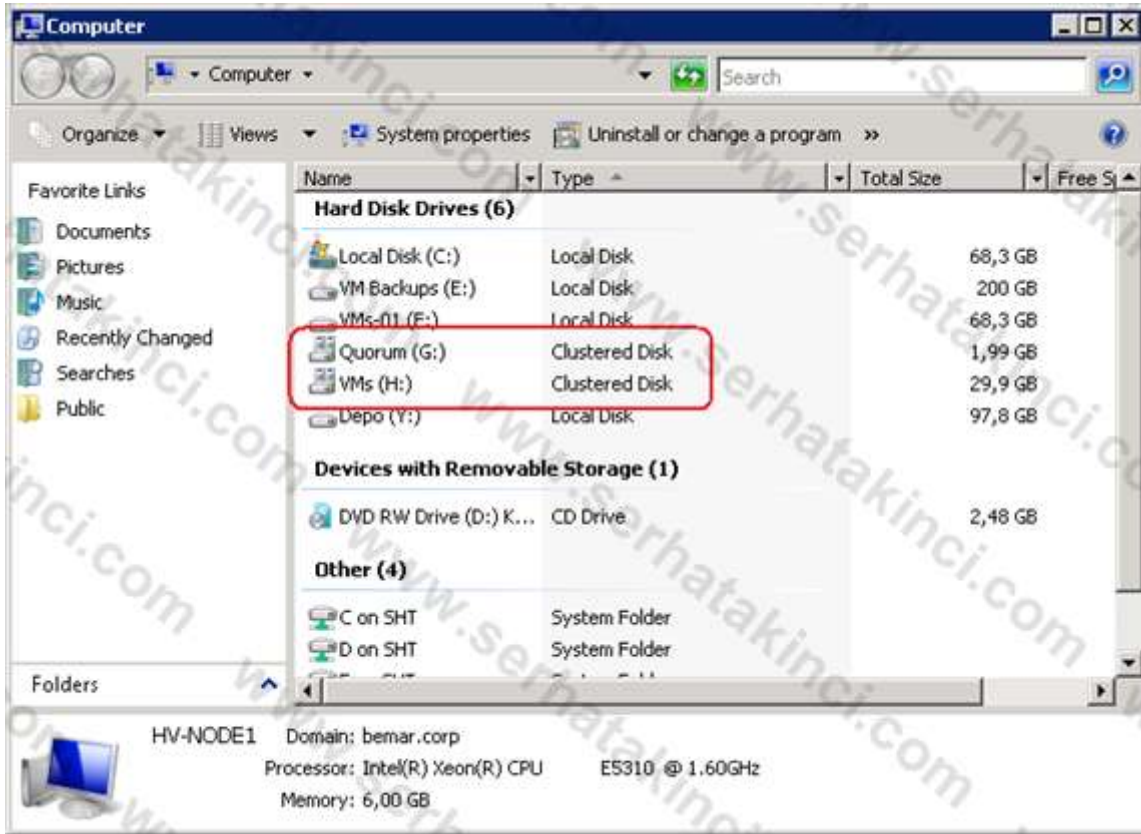


Böylece Virtual Machine servisi ve eklediğimiz bir adet VM için failover cluster yapılandırmasını tamamlanmış olduk.

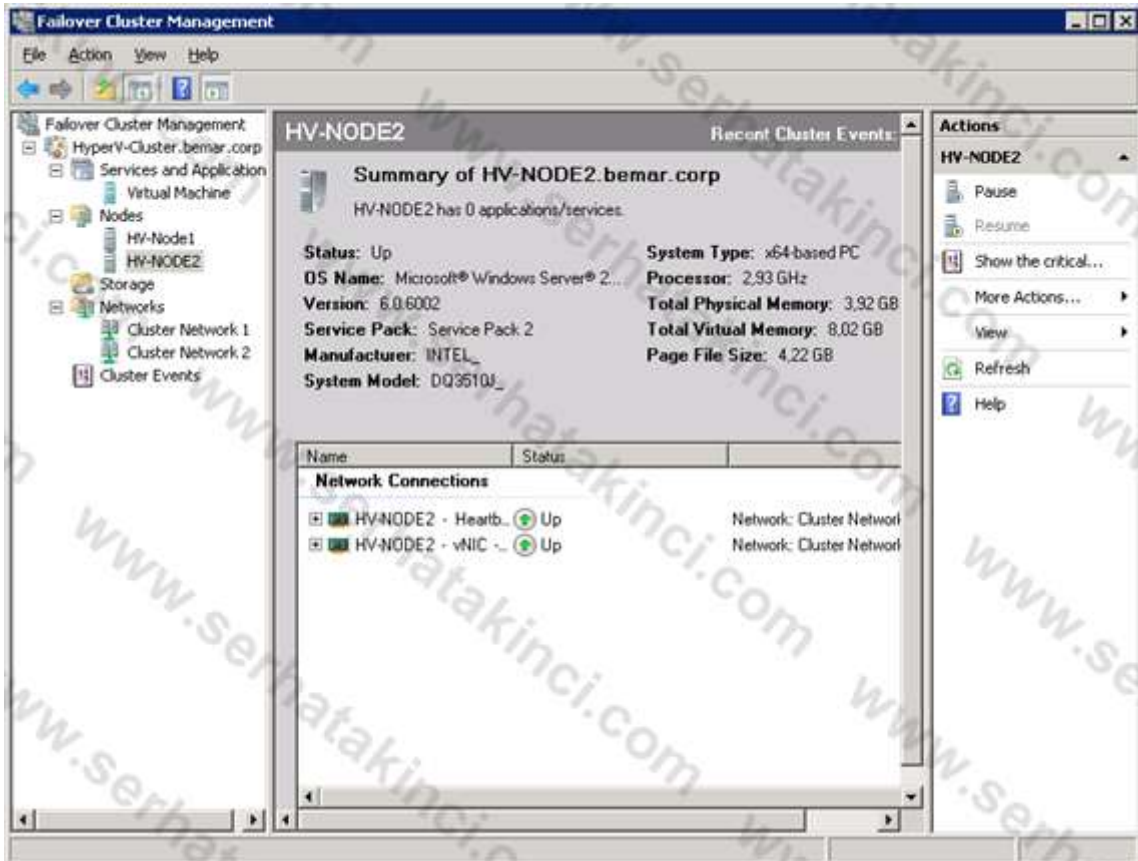
Şu an Cluster olarak çalışan Virtual Machine servisi içerisindeki 1 adet VM'in ve Clustered disklerin Current Owner'ı HV-Node1 makinesi. Bu nedenle HV-Node1'i **aktif node** olarak adlandırıyoruz.



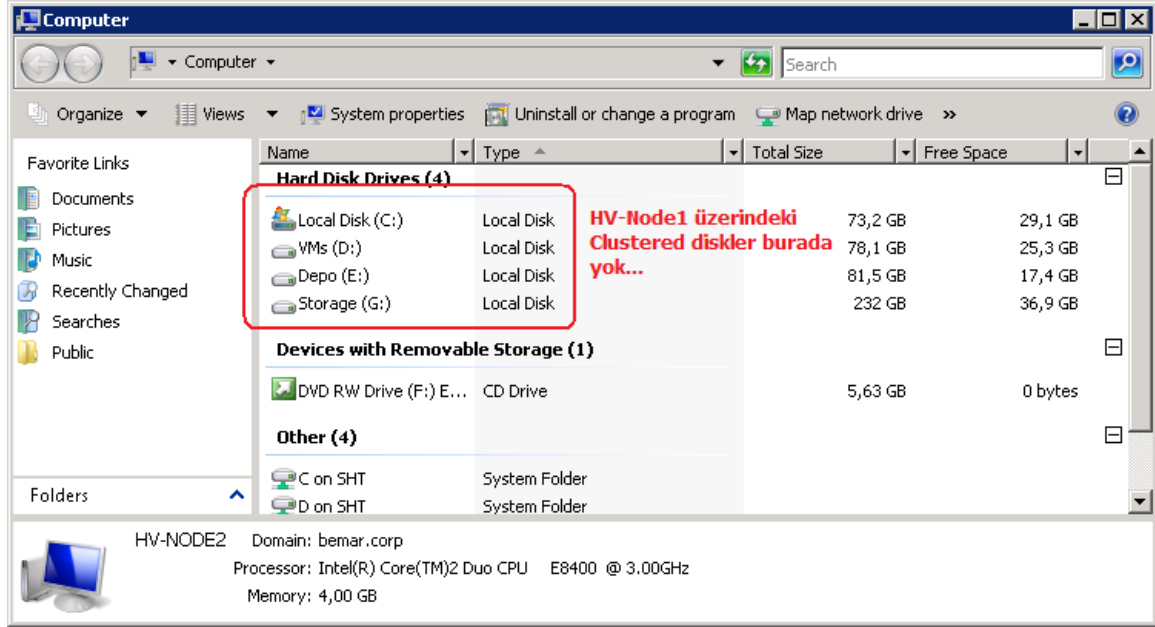
**HV-Node1** üzerinde **Computer** penceresine geldiğinizde ise iki diski erişilebilir şekilde görebilirsiniz.



HV-Node2 ise şu an için herhangi bir cluster kaynağının sahibi değil yani **pasive node** konumunda.



Ayrıca clustered diskler şu an **HV-Node2** üzerinde partition anlamında erişilebilir durumda değil.



Şimdi cluster'ın doğru olarak çalışıp çalışmadığını görmek için bir takım testler yapacağız.

## 10. Hyper-V Failover Cluster Sonrası Gelen Yetenekler

### 10.1. Quick Migration - QM (Hızlı Taşıma)

#### 10.1.1. Quick Migration Nedir?

**Quick Migration** yani hostlar arası **Hızlı Taşıma** özelliği cluster temelli bir özelliktir. Kullanım amacı ise bir fiziksel node üzerinde çalışan sanal makineyi, hızlı bir şekilde diğer fiziksel node üzerine taşımaktır. Yani QM, sanal makine seviyesinde hızlı taşıma sağlıyor.

Konuyu küçük bir senaryo ile örneklendirelim.

Yukarıda bir failover cluster yapısı kurduk ve üzerinde **Cluster-XP** isimli bir VM var. Bu VM şu an **HV-Node1** fiziksel sunucusu üzerinde çalışıyor. Herhangi bir nedenden dolayı HV-Node1'in kapatılması gerekiyor (Örneğin sunucuya fiziksel olarak ram ilavesi yapacağız). Bu durumda **Quick Migration** özelliğini kullanarak birkaç küçük tıklama ile HV-Node1 üzerinde çalışan Cluster-XP isimli VM'i, tüm içeriği ile birlikte cluster'ın diğer üyesi olan HV-Node2 üzerine taşımamız mümkün. İşimiz bittikten sonra HV-Node2 üzerinde çalışan VM'i tekrar Quick Migration ile eski konumuna geri alabiliriz.

Bu özellik **Live Migration** yani **Canlı Taşıma** özelliği ile karıştırılmamalı ki bu sık yapılan hatalardan birisi.

**Quick Migration** ile taşıma sırasında VM network üzerinde hizmet veremez ve taşıma sonrasında hizmet verdiği session'lar kapanmış, connection'lar kesilmiş durumdadır. QM'in en önemli karakteristiği budur, taşıma esnasında VM'e erişim kesilir!

**Quick Migration** aşağıdaki ürünler ile kullanılabilir (cluster üyesi node'lar bu işletim sistemlerinden biri olmalı):

- Windows Server 2008 (Enterprise, Datacenter)
- Hyper-V Server 2008
- Windows Server 2008 R2 (Enterprise, Datacenter)
- Hyper-V Server 2008 R2

**Live Migration** ise tamamen canlı bir taşıma teknolojisidir. Taşıma sırasında VM network üzerinde hizmet verebilir durumdadır ve taşıma süresince üzerindeki servis ve uygulamalara erişim kesilmez. Taşıma tamamlandıktan sonra session'lar ve tüm connection'lar hala bağlı ve açık durumdadır. LM'in en önemli karakteristiği ise taşıma esnasında VM'e ve üzerindeki servislere erişilebiliyor olmasıdır.

**Live Migration** R2 ile gelen bir özelliktir ve aşağıdaki ürünler ile kullanılabilir (cluster üyesi node'lar bu işletim sistemlerinden biri olmalı):

- Windows Server 2008 R2 (Enterprise, Datacenter)
- Hyper-V Server 2008 R2

Quick Migration ile taşıma sırasında VM'e erişim durduğu için taşıma işleminin süresi önem kazanıyor çünkü süre ne kadar uzarsa, VM o kadar hizmet dışı kalacaktır. Peki bu süre ne kadardır ve süreye etki eden faktörler nelerdir?

Aslında bu konuyu fazla detaylandırmak istemiyorum çünkü önümüzdeki makalelerde ayrıntılı olarak ele alacağız. Ama ön bilgi olması açısından şunları söyleyelim.

VM'in memory miktarı başta olmak üzere, node'lar ile storage arasındaki bağlantı tipi ve storage üzerindeki disklerin tipi/hızı Quick Migration işleminin süresine direkt olarak etki eden temel faktörlerdir.

Bunun nedenini anlamak için Quick Migration işleminin background'una bakmak gerekir.

Quick Migration esnasında gerçekleşen işlemler sırası kabaca şöyle.

1. VM saved state duruma alınır.
2. VM durumu (VSV ve BIN bilgisi) diğer node üzerine taşınır.
3. VM diğer node üzerinde start edilir. VSV ve BIN dosyaları sayesinde kaldığı yerden çalışmaya devam eder.

Bir VM için QM (Quick Migration) komutunu verdikten sonra, VM bulunduğu host üzerinde **saved state** dediğimiz bir duruma alınır (VM'in o anki durumu ile dondurulduğunu düşünün). VM'in o anki durumunu (başta sanal memory içeriğini ve process bilgilerini) tutan dosyalar ise VSV ve BIN dosyalarıdır. QM işleminin süresine etki eden temel faktör bu dosyaların boyutlarıdır. Bu dosyaların boyutları ise VM'e atadığımız sanal memory miktarı ile doğru orantılıdır.

Örneğin 256MB memory'e sahip bir VM için yaklaşık 256mb boyutunda bir BIN dosyası vardır ve bunun QM sırasında save edilmesi, bilgisinin diğer node üzerine taşınması gerekir.

Ama 2GB memory'e sahip bir VM için 2048mb boyutunda bir BIN dosyası vardır ve haliyle taşıma işlemi daha uzun sürecektir.

Bunun dışında storage ve node'lar arasındaki bağlantı teknolojisi de bu taşıma işleminin hızına direkt olarak etki edecektir.

Dediğim gibi ayrıntılar daha sonra :)

Bizim kurmuş olduğumuz iSCSI cluster yapısında 1GB ram'e sahip makineyi 10sn gibi bir sürede QM ile diğer node'a taşıyabiliyoruz.

Aşağıdaki tablo ise storage bağlantı teknolojileri ve VM üzerindeki ram miktarına göre bazı ortalama rakamları yansıtmaktadır. iSCSI tarafındaki rakamlara takılmayın, süreler çok daha aşağılara çekilebiliyor.



VM Memory	1 GbE iSCSI	2 Gb FC	4 Gb FC
512 MB	~8 seconds	~4 seconds	~2 seconds
1 GB	~16 seconds	~8 seconds	~4 seconds
2 GB	~32 seconds	~16 seconds	~8 seconds
4 GB	~64 seconds	~32 seconds	~16 seconds
8 GB	~2 minutes	~64 seconds	~32 seconds

Quick Migration işlemi biz yöneticiler tarafından başlatılır ve geçiş süresince VM'in erişilemez durumda olacağı bilinir. Bu nedenle QM'i **planlı kesintiler** sınıfına sokabiliriz.

Şimdi kurmuş olduğumuz failover cluster yapısında QM (Quick Migration)'i nasıl yapıyoruz bir bakalım. Hem de yapıyı test etmiş olalım.

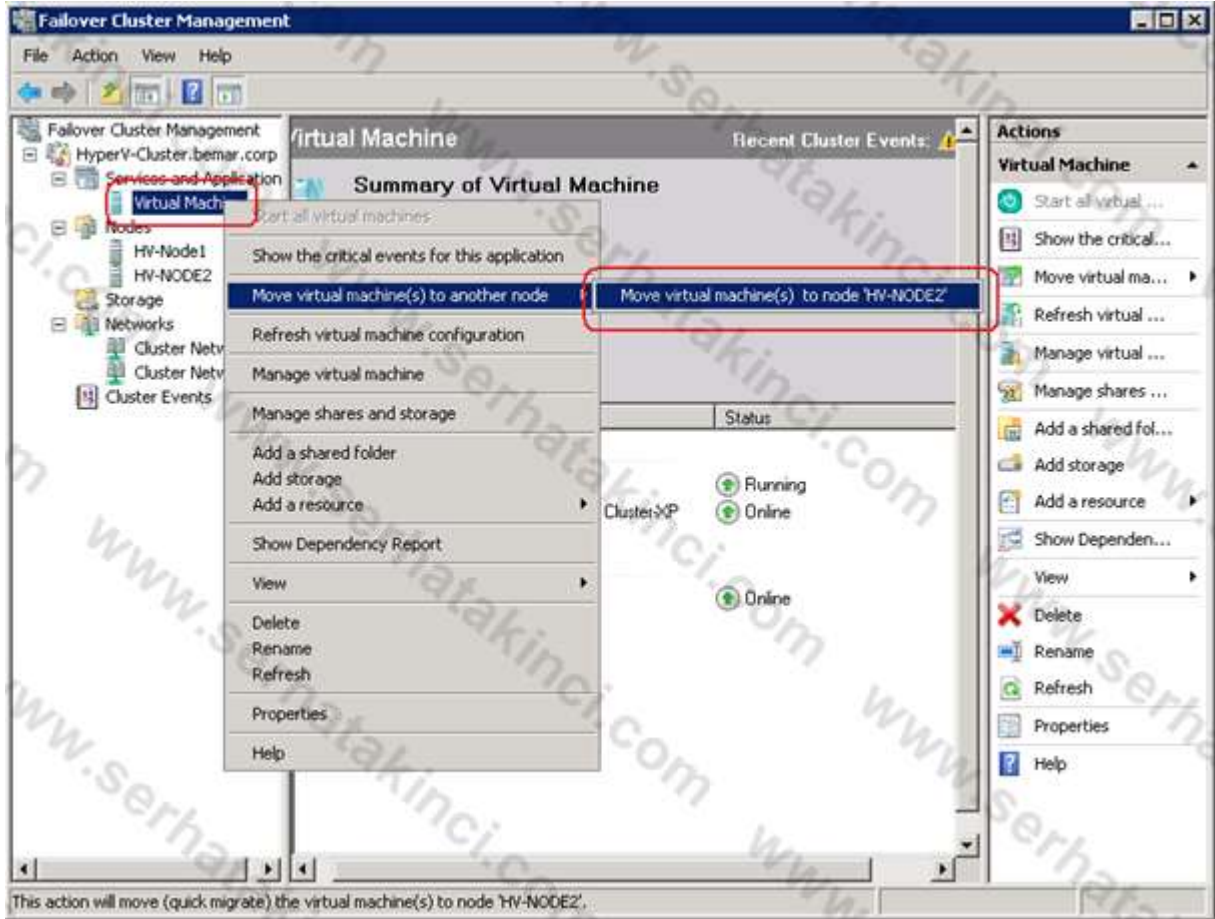
### 10.1.2. Quick Migration Nasıl Yapılır?

**HV-Node1** üzerinde **Failover Cluster Management** konsolunu açıyoruz.

QM işlemini **Failover Cluster** konsolu üzerinden yapabileceğimiz gibi **SC Virtual Machine Manager 2008** ile de yapabiliriz.

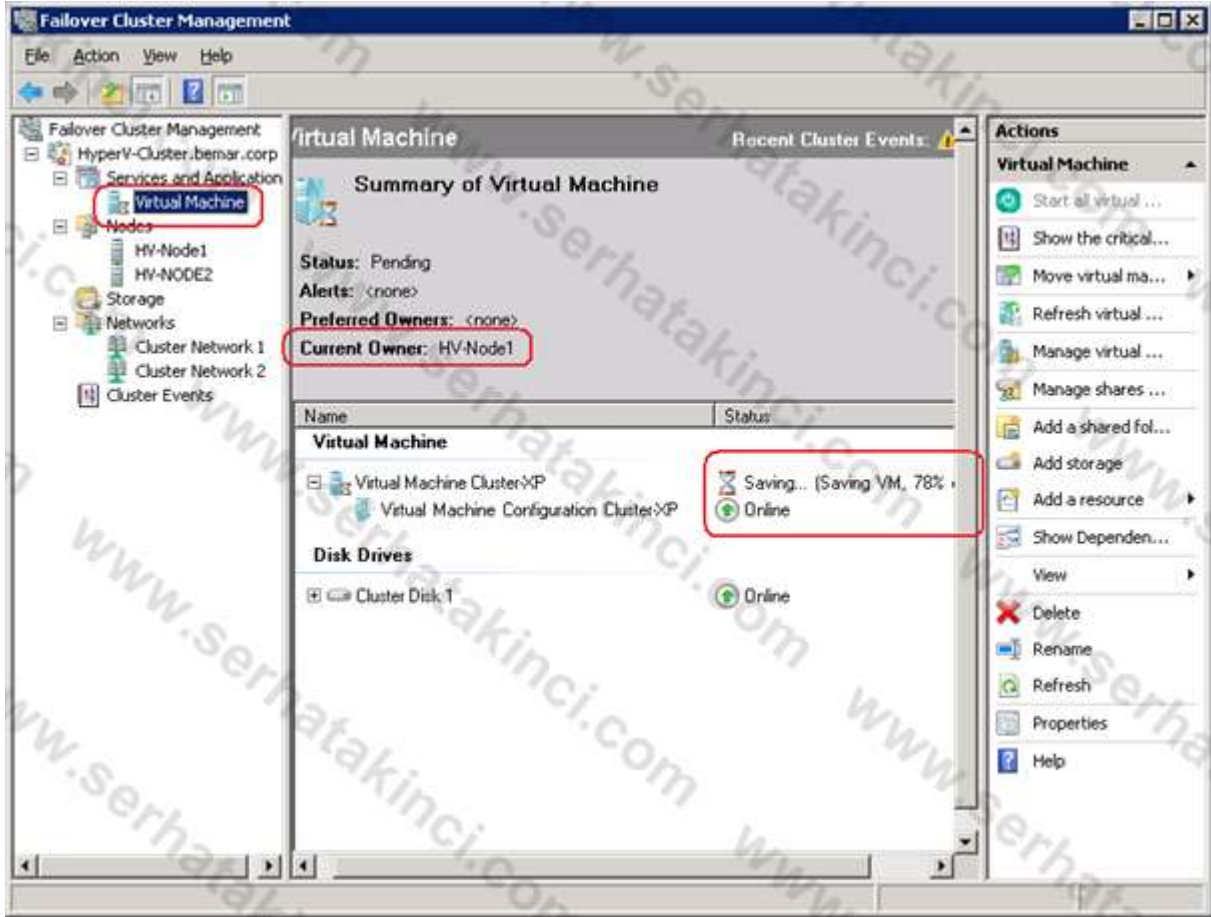
**Services and Applications** altındaki **Virtual Machine** servisine sağ tıklıyoruz ve **Move virtual machine(s) to another node** altındaki **Move virtual machine(s) to node 'HV-Node2'** seçiyoruz.

Bu komut ile Virtual Machine servisi içerisinde eklemiş olduğumuz ve şu an **HV-Node1** tarafından sahiplenilmiş olan VM'i/VM'leri, **HV-Node2** üzerine QM yapıyoruz (biz bir adet eklemiştik: Cluster-XP).



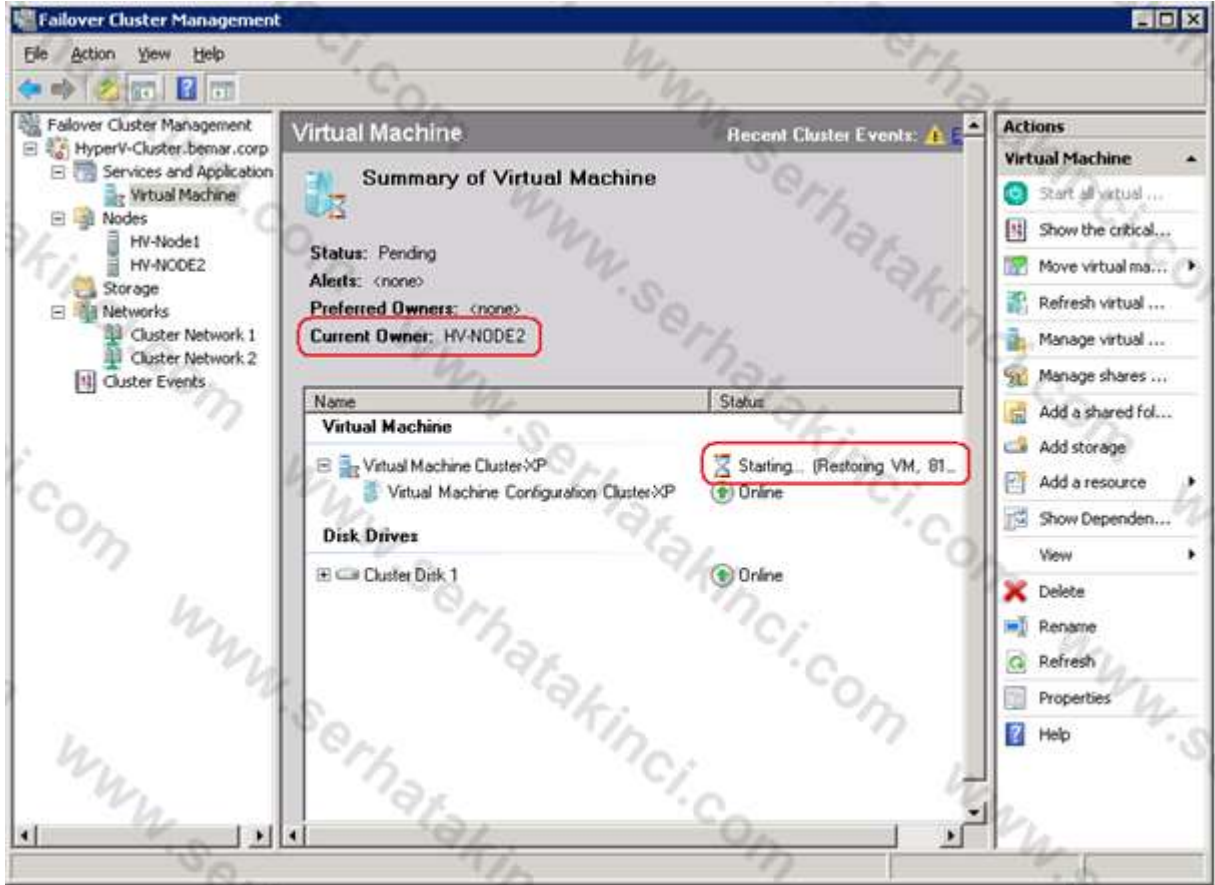
Komutu verdikten sonra işlem hemen başlıyor. Aşağıda da gördüğümüz gibi öncelikle VM save ediliyor yani **saved state** durumuna geçiyor.

Bu esnada ilgili VM'in sahibi (Current Owner) HV-Node1. VM HV-Node1 üzerinde çalışıyor. Bunu da aşağıda görebiliyoruz.

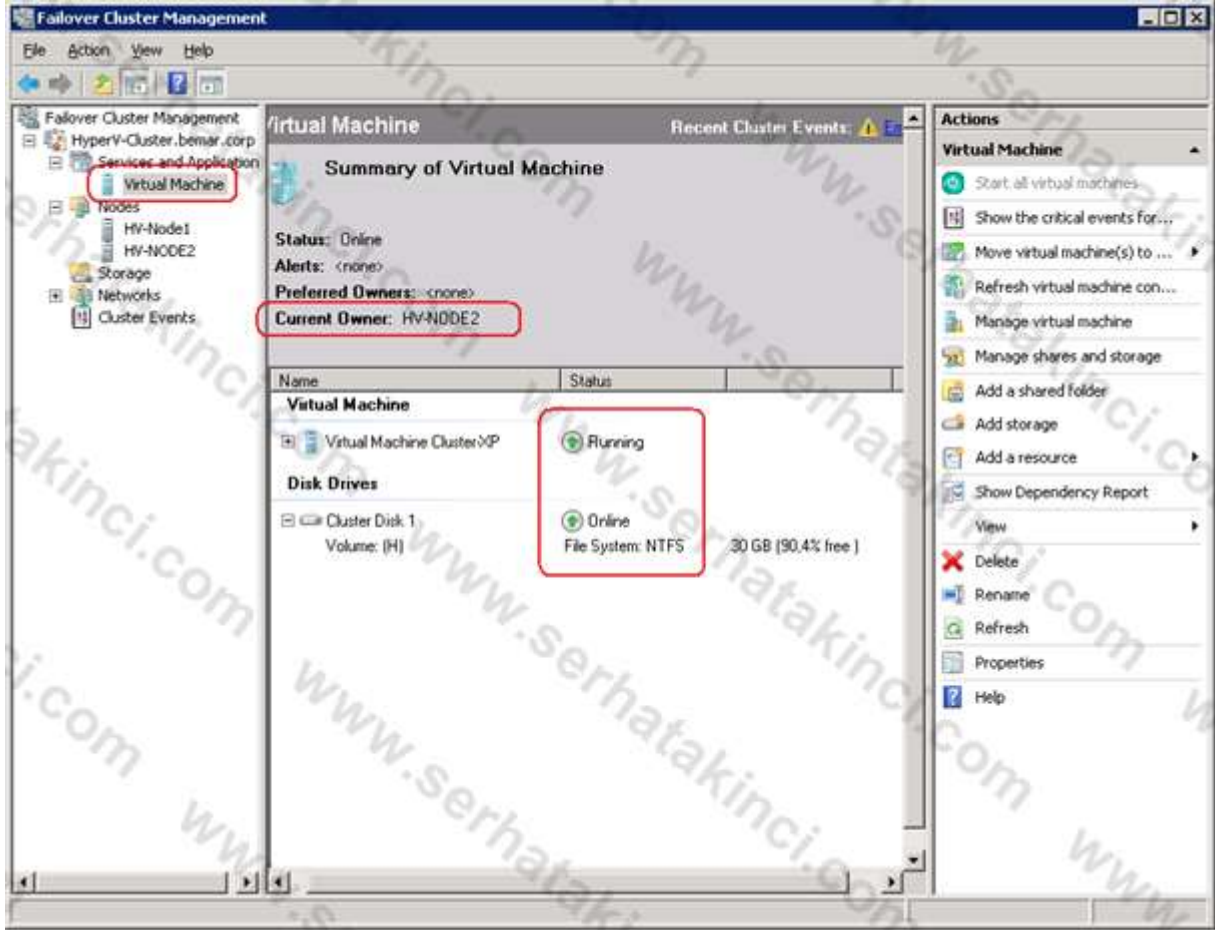


**Saving** işlemi saniyeler içinde tamamlandıktan sonra ilgili VM artık diğer node'a (bizim yapımızda HV-Node2) devrediliyor ve **owner**'lık değişiyor.

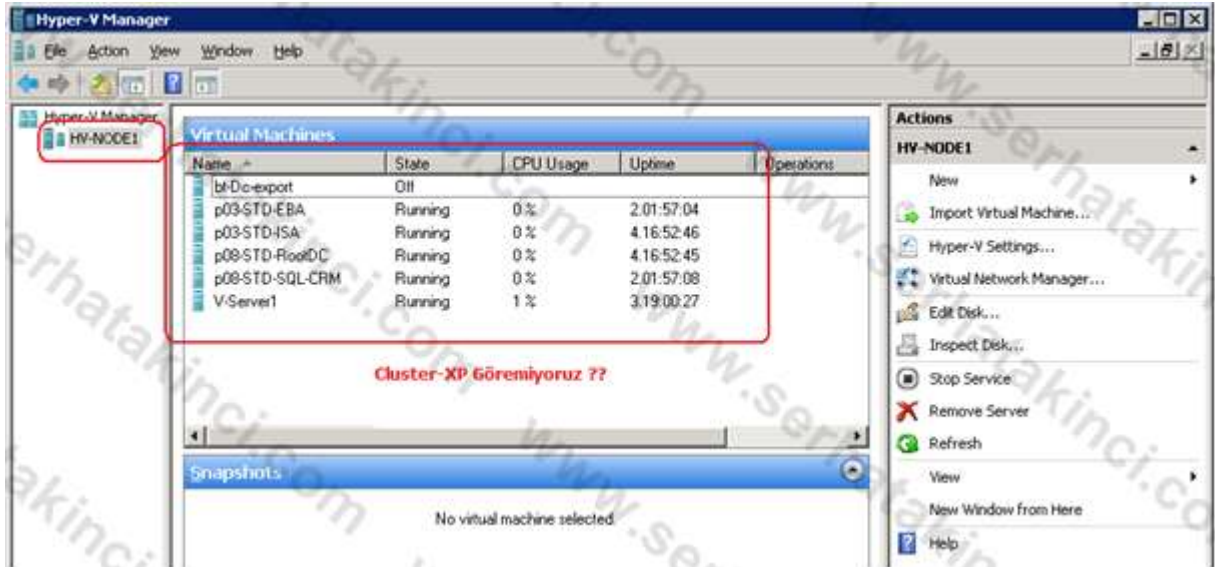
VM HV-Node2 üzerinde start olmaya başladı. Artık Current Owner HV-Node2.



Kısa bir süre içerisinde işlem tamamlanıyor ve Cluster-XP isimli makine diğer HV-Node2 üzerinde çalışmaya devam ediyor.

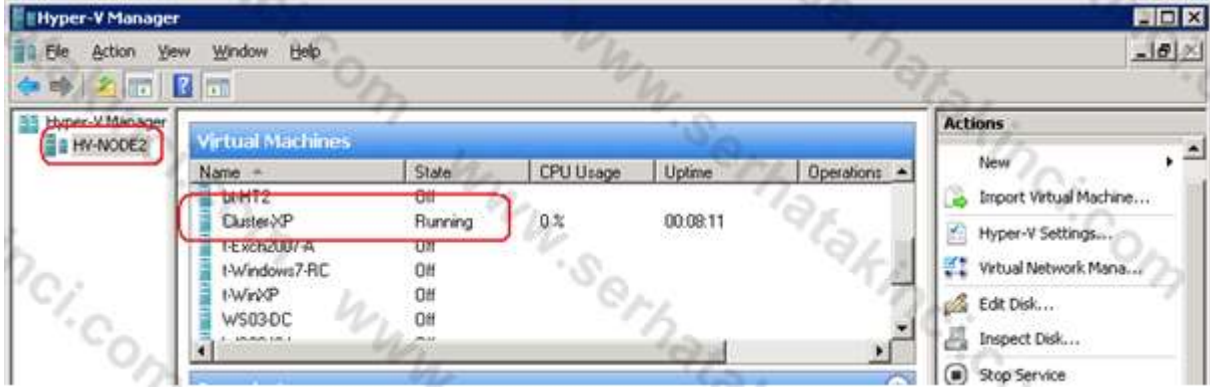


Bu noktada HV-Node1 yani Cluster-XP'nin eski sahibi olan node üzerinde Hyper-V Manager konsolunu açarsanız, Cluster-XP'nin yer almadığını görebilirsiniz. Çünkü artık diğer node üzerinde.



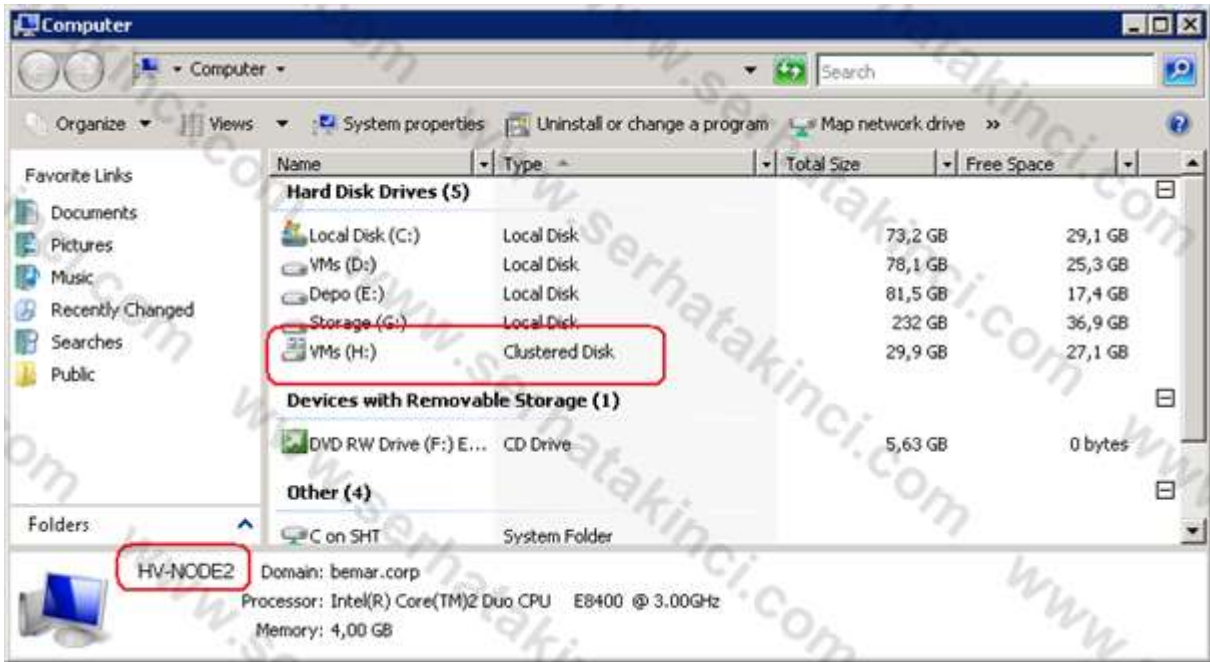
HV-Node2 üzerinde ise Hyper-V Manager konsolunu açtığınızda Cluster-XP'nin yer aldığını görebilirsiniz.





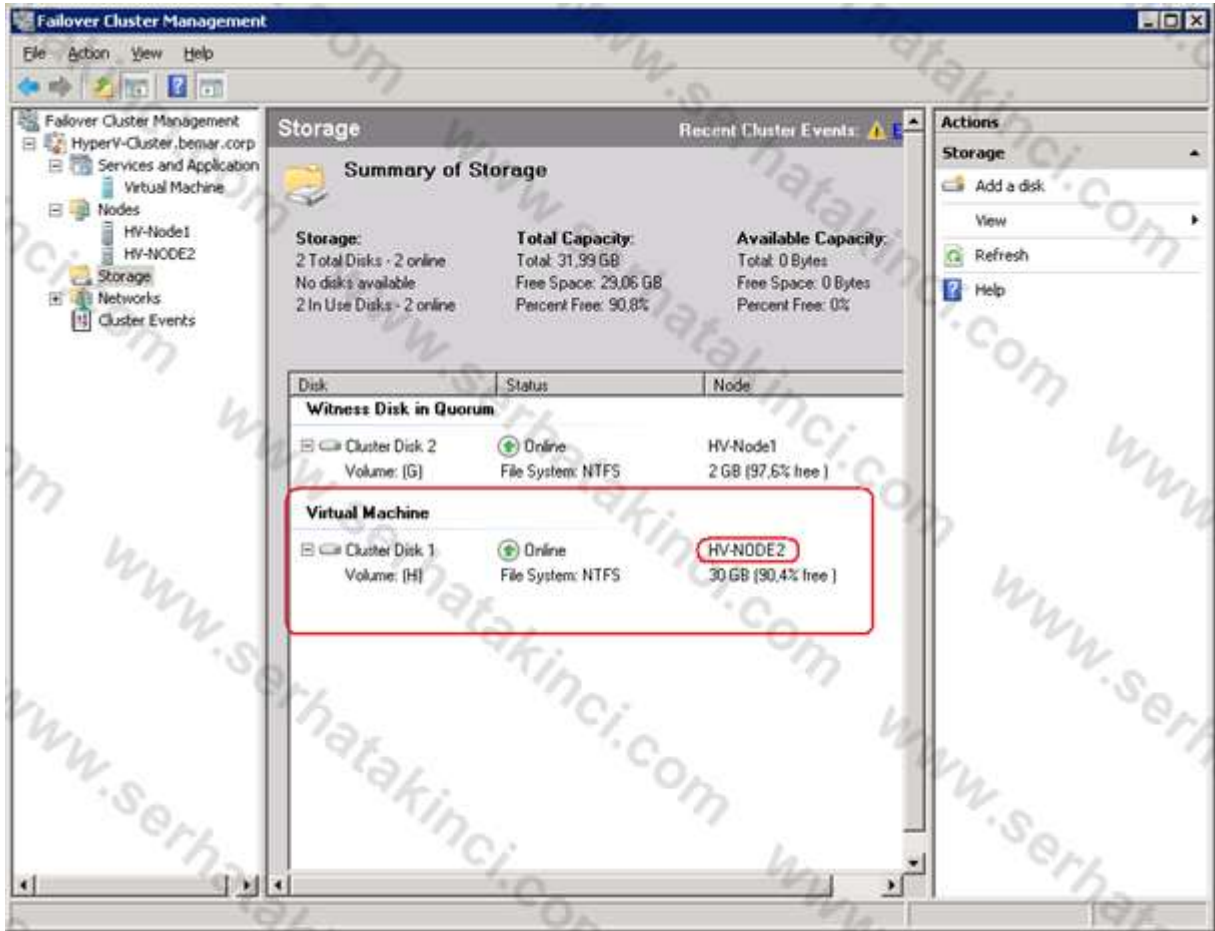
Hatırlarsanız Cluster-XP VM'i HV-Node1 üzerinde çalışırken, clustered diskler de HV-Node1 üzerinde direkt olarak erişilebilir durumdaydı.

Şu an ise clustered diskler den birisi (VM'in yer aldığı disk) otomatik olarak HV-Node2 üzerinde erişilebilir duruma geçti.



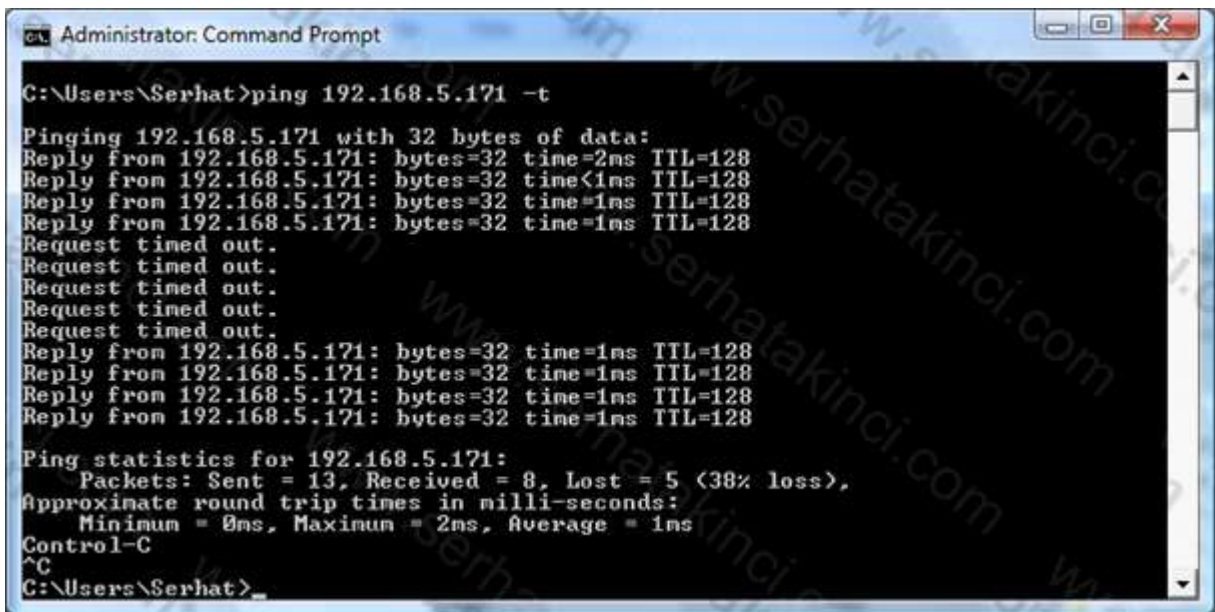
Diğer clustered disk yani quorum bilgisinin tutan disk ise hala HV-Node1 üzerinde çalışıyor çünkü HV-Node1 up durumda. QM işleminde quorum diskin yeri değişmez.

Disk sahiplik durumlarını **Failover Cluster Management** konsolunda **Storage** bölümünde daha net görebilirsiniz.



Quick Migration başlatmadan hemen önce network üzerinden Cluster-XP sanal makinesine ping atmaya başlamıştım.

QM bitene kadar olan ping istatistiği aşağıdaki gibidir.

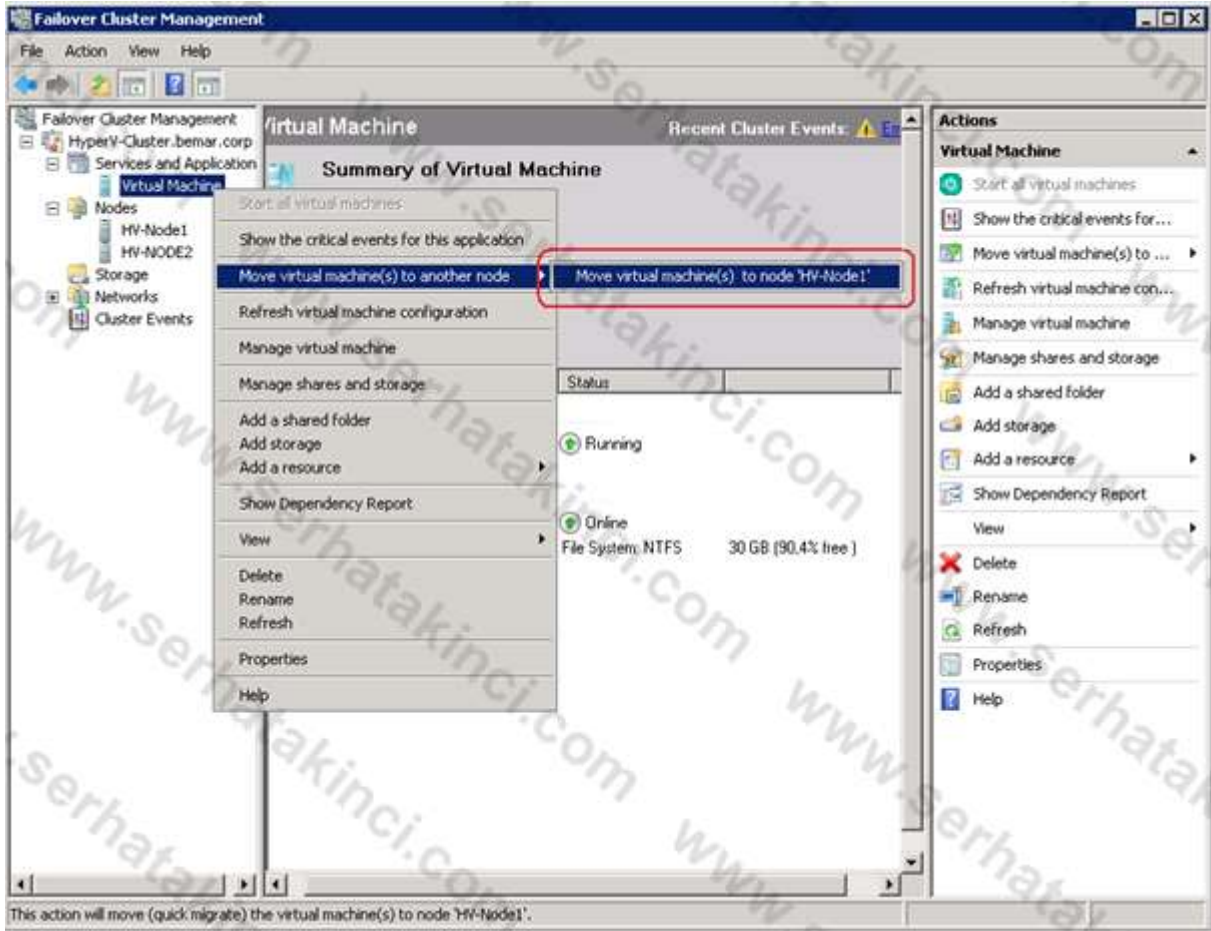


Görüldüğü gibi 5 ping paketi kaybı sonrasında VM tekrar network üzerinde erişilebilir hale gelmiş.

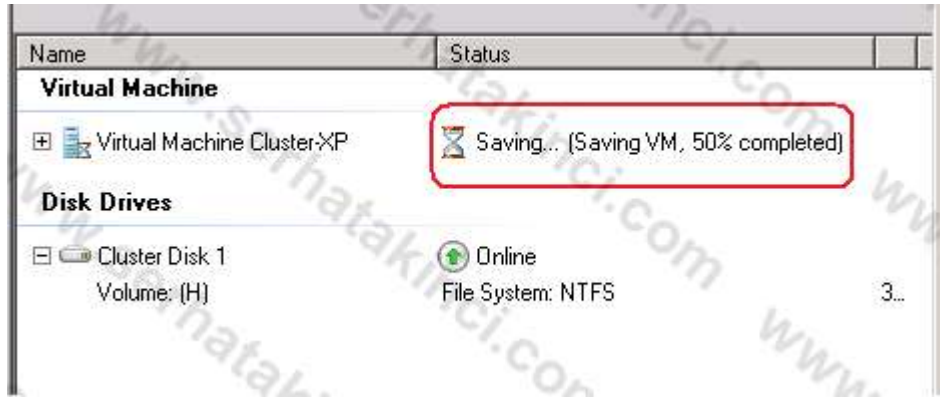
Şimdi VM'i tekrar eski yerine taşıyalım. Yani HV-Node2 üzerinden yine HV-Node1 üzerine alalım.

Yine **Failover Cluster Management** konsolu üzerinde **Services and Applications** altındaki **Virtual Machine** servisine sağ tıklıyoruz ve **Move virtual machine(s) to another node** altındaki **Move virtual machine(s) to node 'HV-Node1'** seçiyoruz.

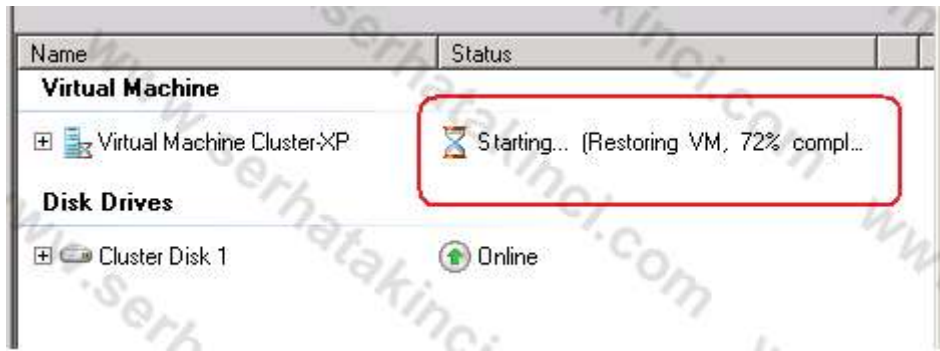
Eğer cluster yapımızda başka node'lar olsaydı onları da burada görüyor olacaktık.



Bu komut sonrası yine VM hızlıca save ediliyor.

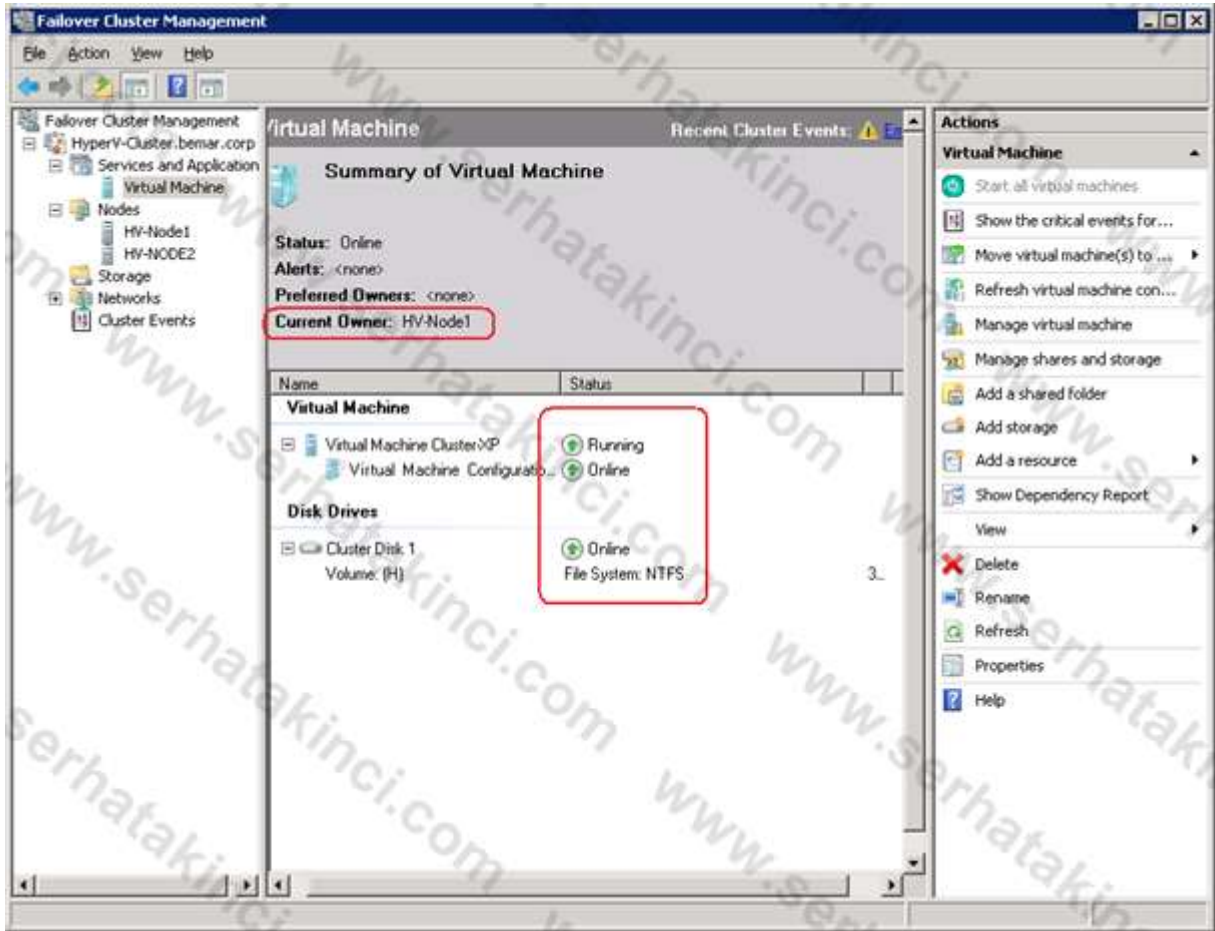


Ve HV-Node1 üzerinde tekrar start ediliyor.



Taşıma işlemi bittiğinde current owner olarak HV-Node1 görünüyor ve VM online/running durumda.





**Quick Migration** işlemi bu kadar. Gördüğünüz gibi birkaç küçük tıklama ile VM'leri cluster üyesi sunucular arasında rahatlıkla dolaştırabiliyoruz :)

## 10.2. VM Failover (Sanal Makine Hata Telafisi)

### 10.2.1. VM Failover Nedir?

Cluster yapısına eklenmiş Host ve VM'ler için failover yeteneği temelde bir HA (High Availability) özelliğidir.

Cluster üyesi Node'lardan birinin başına bir şey geldiğinde (donanımsal bir başarısızlık nedeni ile sunucunun down olması gibi), down olan node üzerindeki servis ve uygulamanın (bizim yapımızda Virtual Machine servisi) up durumda olan diğer node üzerinde hizmet vermeye devam etmesi durumudur.

Yine küçük bir senaryo ile örneklendirelim.

Yapımızdaki Cluster-XP makinesi şu an HV-Node1 üzerinde çalışıyor. Diğer cluster üyesi node ise HV-Node2 ve şu an pasive node konumunda bekliyor.

Gün içerisinde HV-Node1 üzerinde bir donanım arızası oluşuyor ve sunucu bir anda down oluyor. Kısa bir süre içerisinde telefonlar yağmaya başlıyor çünkü HV-Node1



üzerinde hizmet veren VM ve servislere erişilemiyor ve bu hizmetleri kullanan kullanıcılarımız hemen telefona sarılıyor :) )

Hatırlarsanız makalenin ilk bölümlerinde heartbeat isimli bir network ten bahsetmiştik. Bu network için farklı bir ip config yaparak public networkten ayırmış ve cluster üyesi sunucuların bu network üzerinden birbirlerini yokladıklarını, up durumda olup olmadıklarını kontrol ettiklerini söylemiştik. İşte bu network üzerinden yapılan kontroller sırasında down olmuş bir node tespit edilirse, otomatik olarak gerekli aksiyon alınabiliyor.

Senaryomuza dönersek, HV-Node1 down olduğu zaman HV-Node2 bunu kısa süre içerisinde fark edebiliyor çünkü sunucular heartbeat network üzerinden sürekli birbirlerini kontrol ediyor. Daha sonra HV-Node2 gerekli cluster kaynaklarını ve HV-Node1 üzerinde çalışan servisleri otomatik olarak kendi üzerine alıyor yani VM'ler otomatik olarak HV-Node2 üzerinde failover oluyor (hata telafisi).

Bu noktada geçen süre, node'un down olduğuna karar verilmesi ve ilgili kaynakların diğer node üzerine taşınması gibi işlemler ile birlikte ortalama 1-2 dakikadır (değişkenlik gösterir). Node'lar arası kontrol süresini çok kısa tutmak sağlıklı değildir çünkü heartbeat network üzerinde meydana gelebilecek anlık ve ya kısa süreli geçici problemlerde node'lar birbirine ulaşamaz ise ortalık bir anda karışabilir :) )

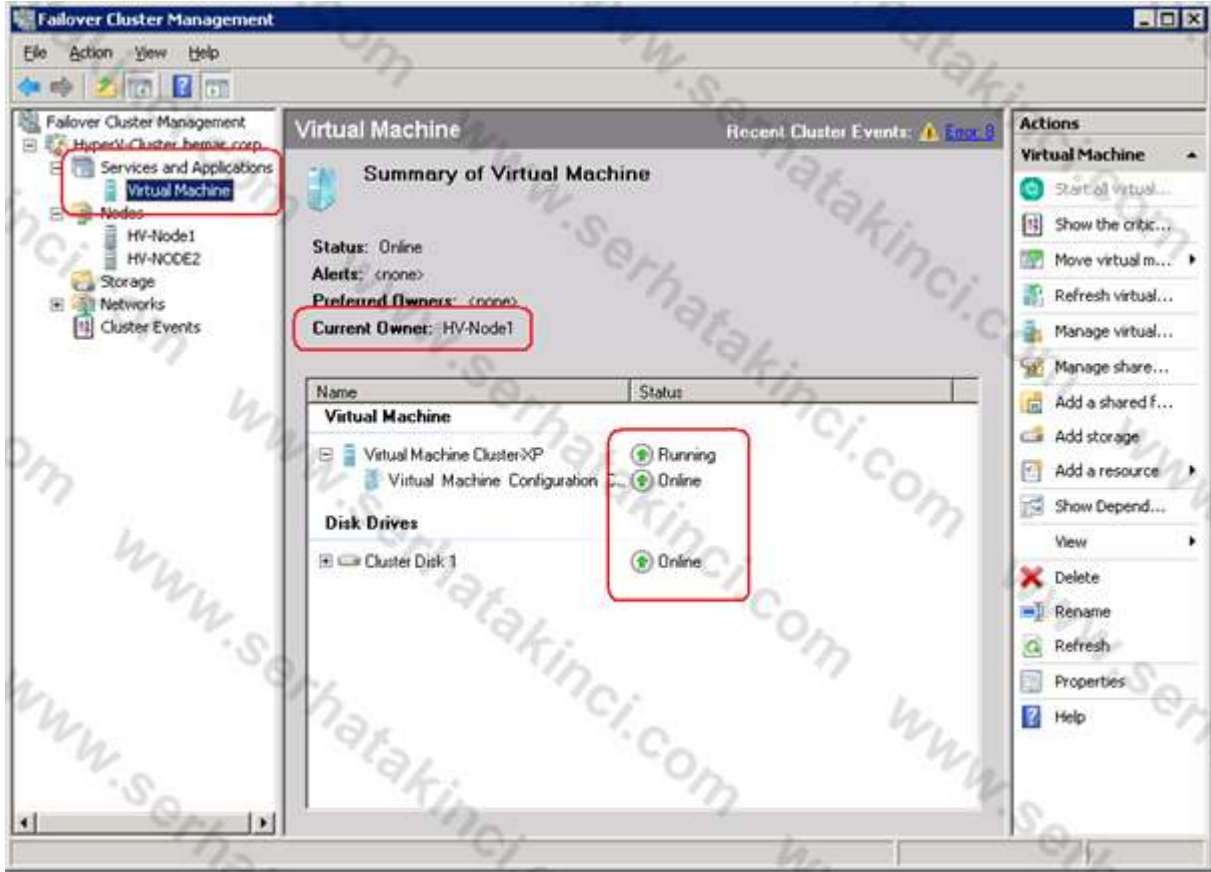
Bir sunucunun down olması durumu biz yöneticilerin isteği dışında gerçekleşen bir olay olduğu için **plansız kesintiler** sınıfına sokuyoruz.

Şimdi failover aksiyonunu test edelim.

### 10.2.2. VM Failover Nasıl Yapılır?

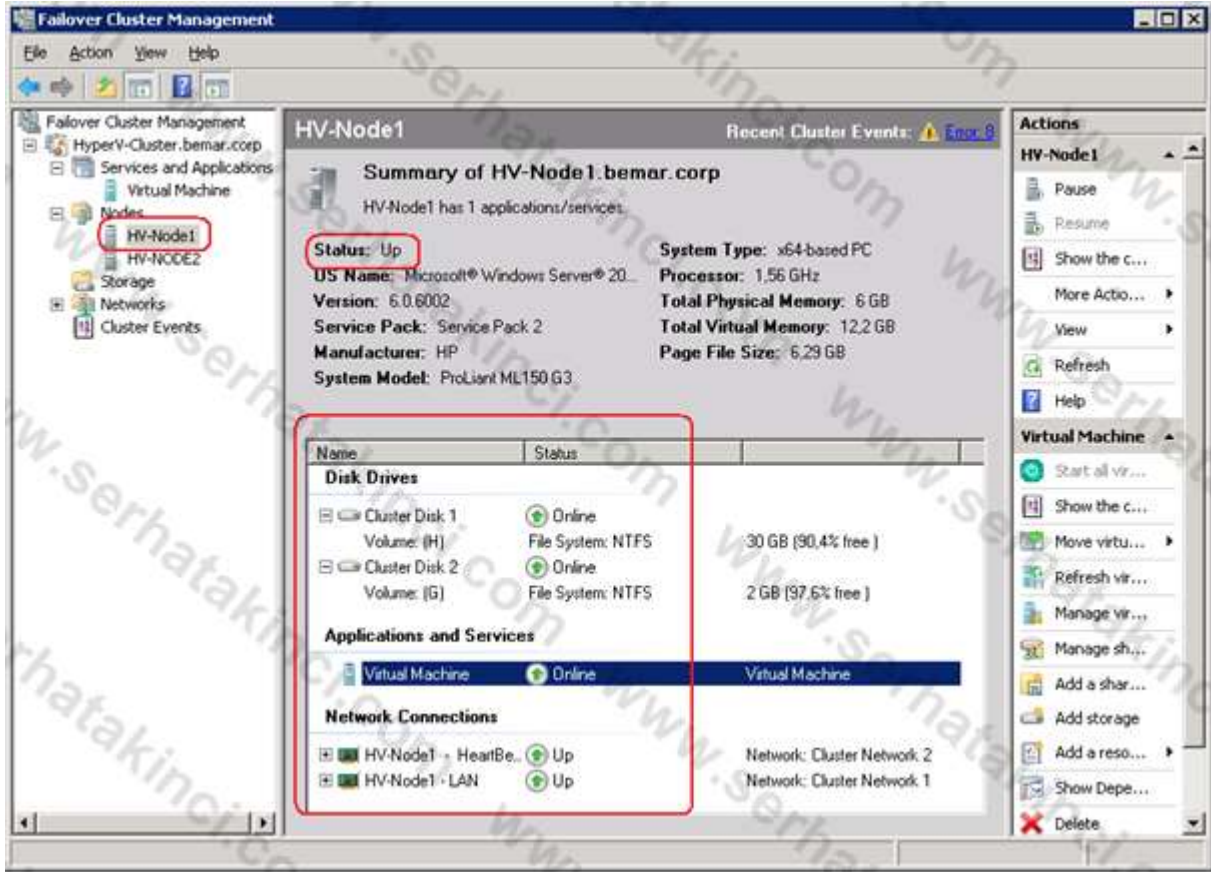
Yapılması gereken ekstra bir ayar yok. Daha önceden kurmuş olduğumuz failover cluster yapısı dahilinde bu özellik otomatik olarak etkin durumdadır. Cluster yapısına **Services and Applications** olarak **Virtual Machine** servisinin eklenmesi ve ilgili VM'lerin seçilmesi yeterli (biz bunu zaten yapmıştık).

HV-Node1 üzerinde Failover Cluster Management konsolunu açtığımızda Virtual Machine servisinin sahibinin HV-Node1 olduğunu görebiliyoruz ve servis şu an online durumda.



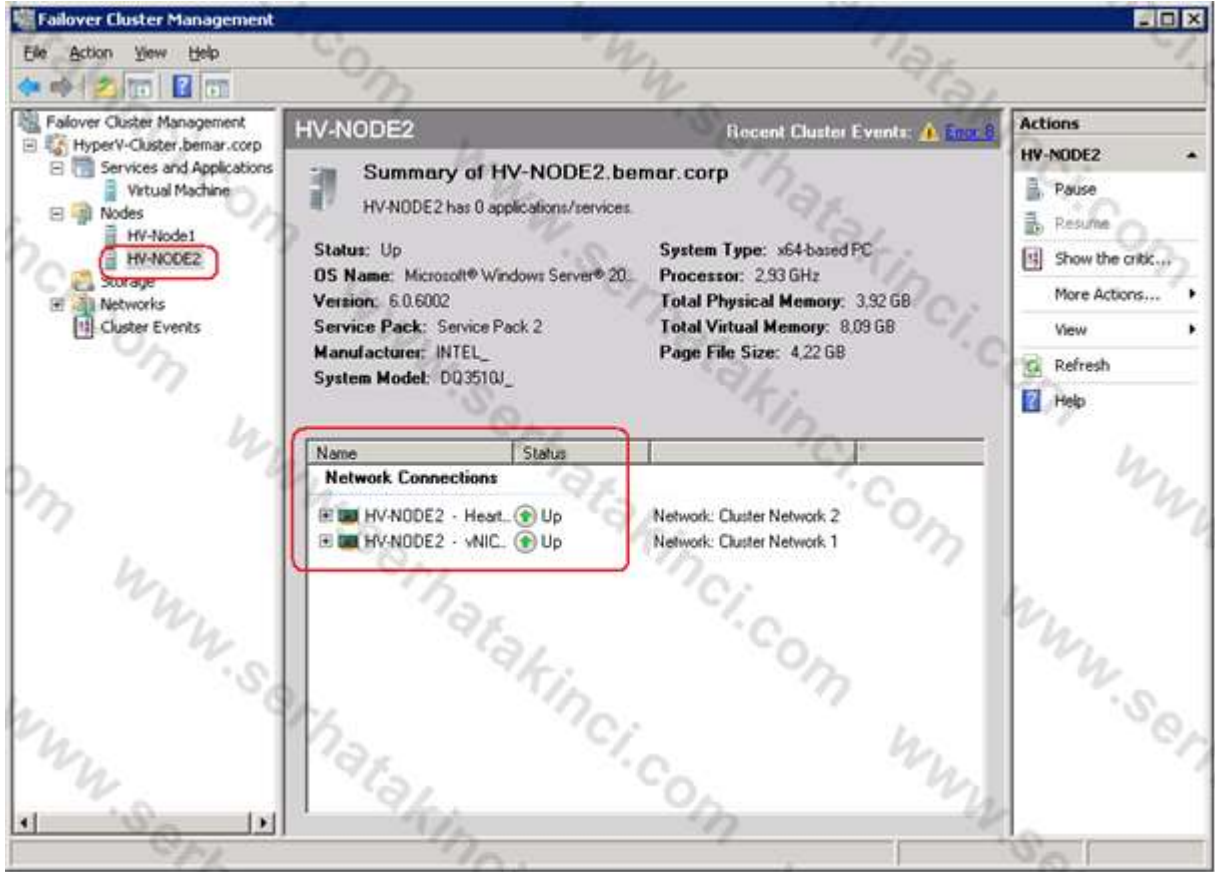
**Nodes** altında HV-Node1'e baktığımızda her iki clustered disk'in ve virtual machine servisinin HV-Node1 üzerinde olduğunu açıkça görebiliyoruz. Yani **aktif node** durumunda.

Ayrıca node **up** durumda.



Şu an HV-Node2 ye baktığımızda ise üzerinde hiçbir cluster kaynağının olmadığını görüyoruz. Yani **passive node** durumunda.

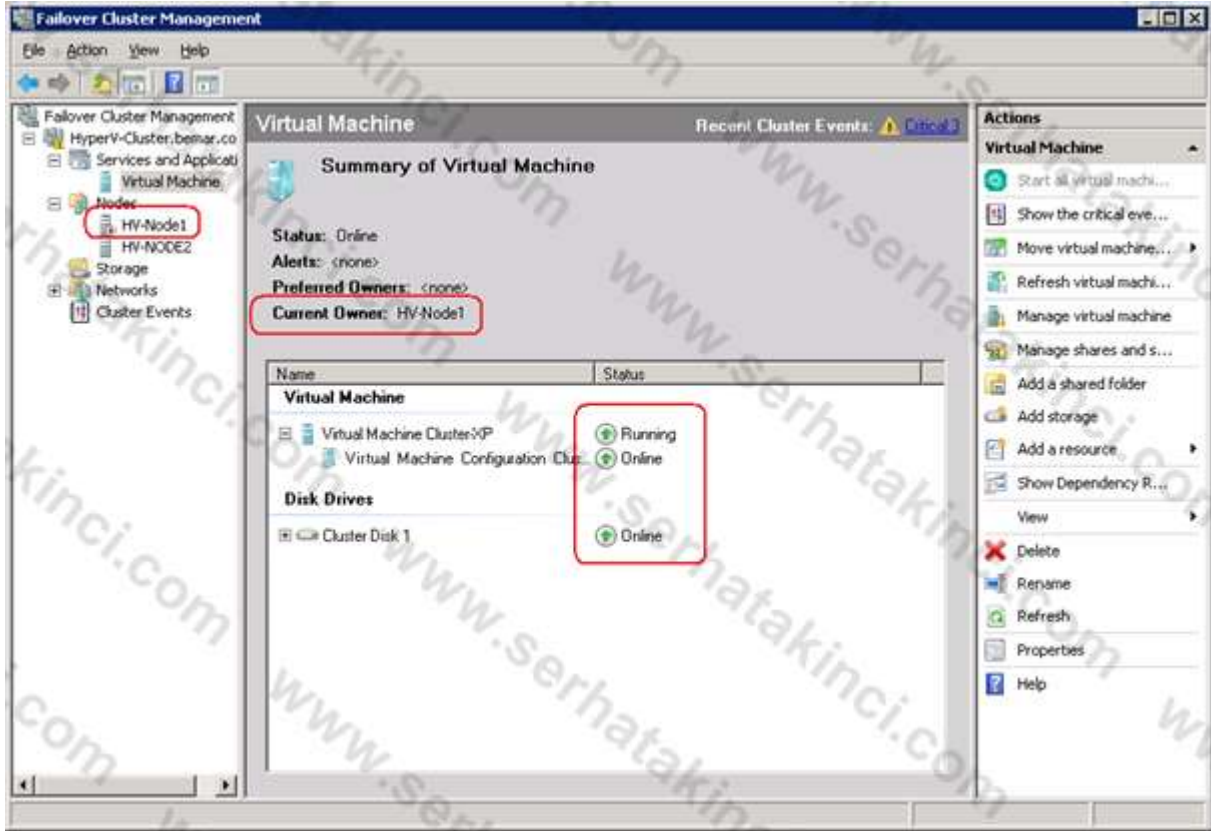
Ayrıca o da **up** durumda.



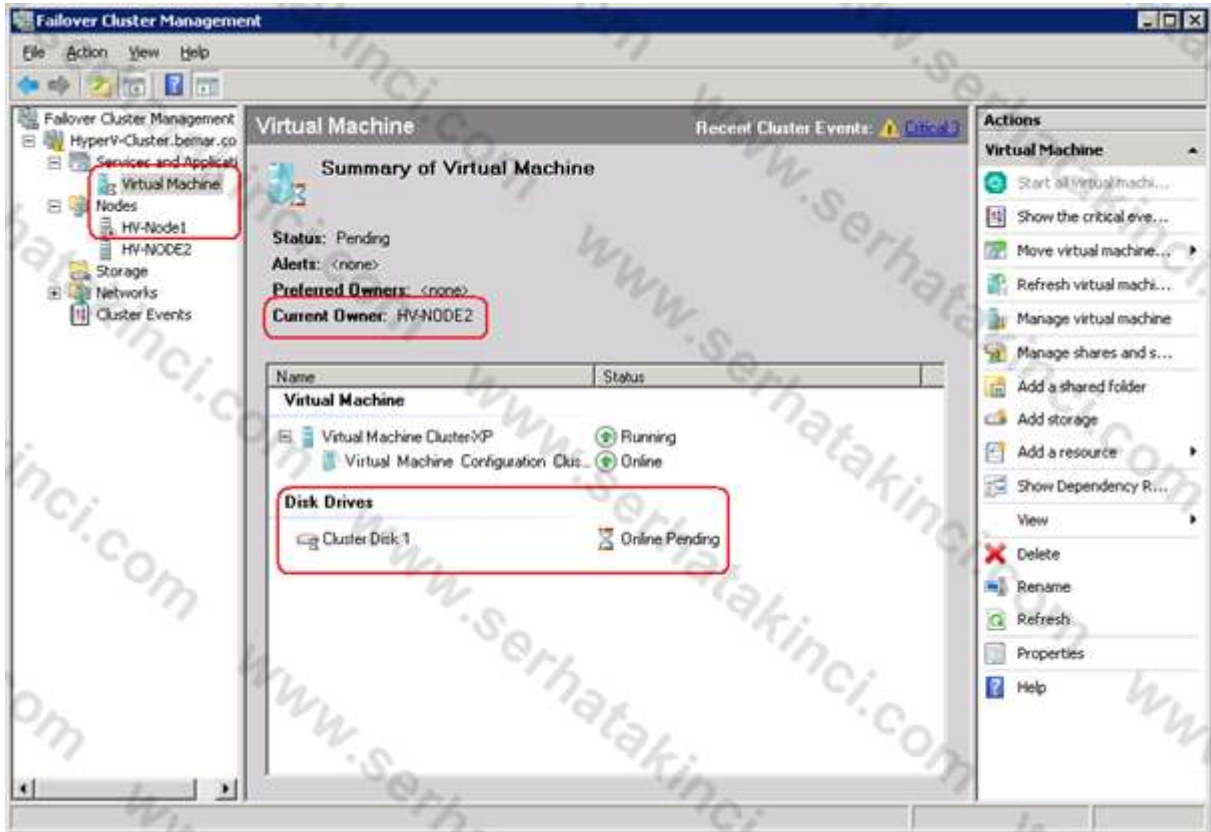
Şimdi herhangi bir nedenden dolayı HV-Node1'in down olduğunu düşünelim. Ben test için HV-Node1'in fişini çekiyorum :)

Cluster ortamı hemen HV-Node1 de bir problem olduğunu anlıyor. Aşağıdaki görüntüde HV-Node-1 üzerindeki küçük kırmızı uyarıyı görebilirsiniz.

Ayrıca şu an Virtual Machine servisinin sahibi hala HV-Node1.

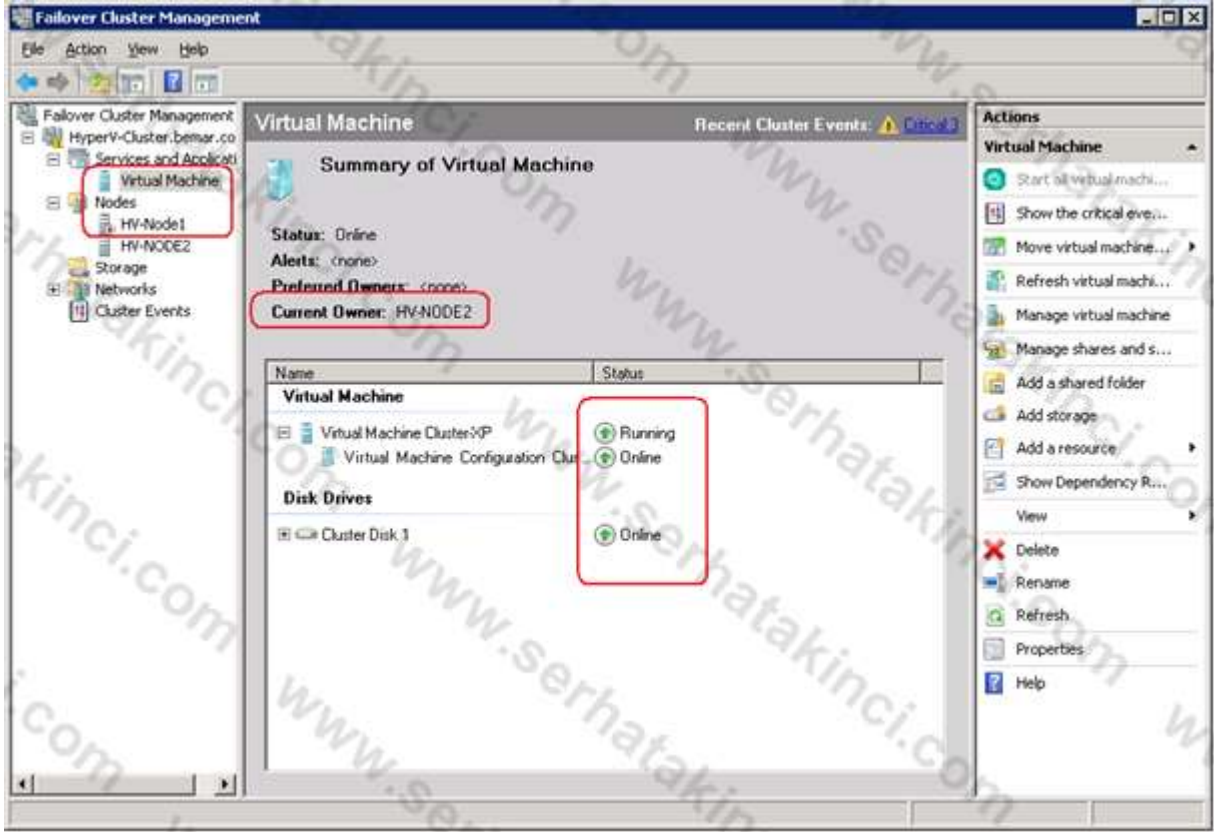


Saniyeler içerisinde servis **pending** duruma geçiyor ve HV-Node2'ye aktarılıyor.

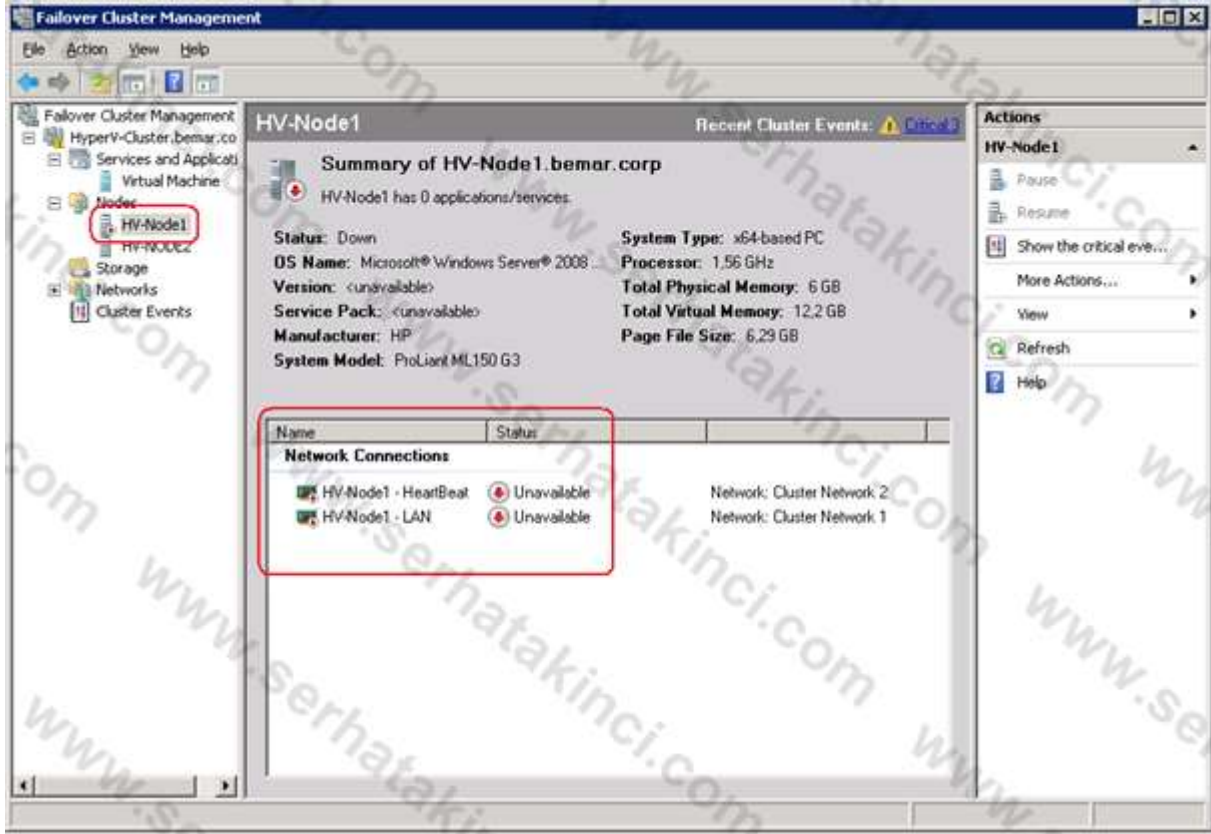




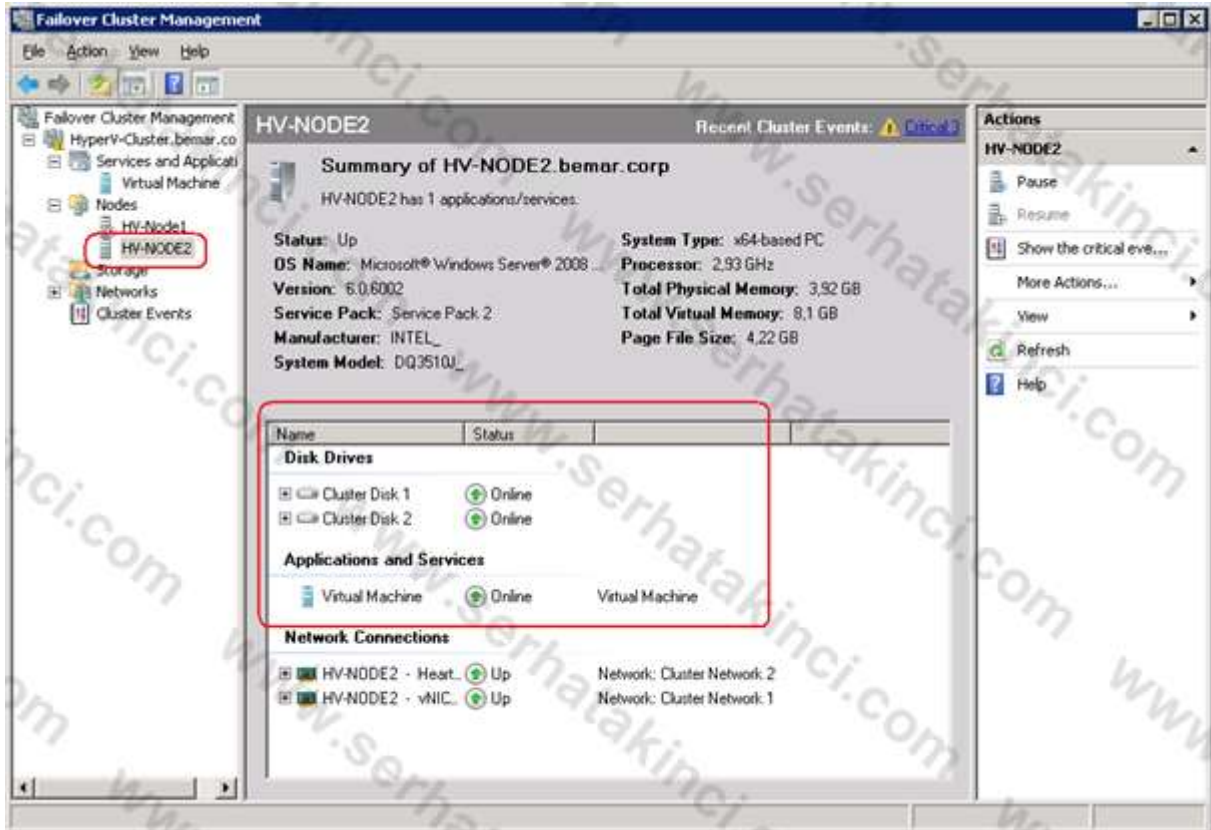
İşlem tamamlandığında servisin yeni sahibi HV-Node2 ve Cluster-XP online olmuş durumda.



HV-Node1 şu an kapalı durumda bekliyor ve üzerinde hiçbir cluster kaynağı olmadığını görebiliyorsunuz.



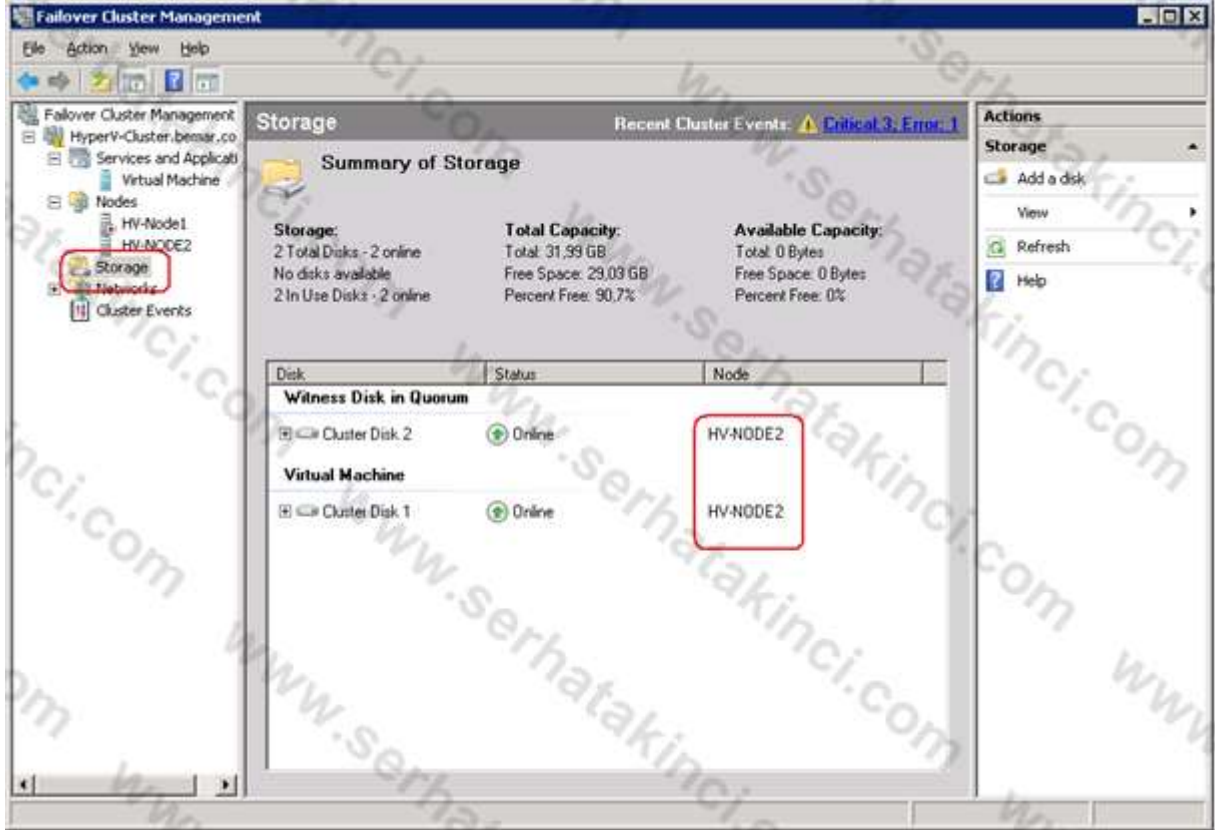
HV-Node2 ise tüm cluster kaynaklarını kendi üzerine almış durumda ve sistem çalışmaya devam ediyor.



Storage bölümünden baktığımızda ise iki diskin de HV-Node2 tarafından sahiplenildiğini görüyoruz.

QM işlemini hatırlayın, sadece VM'in olduğu diskin sahipliği değişiyordu çünkü planlı bir kesinti idi ve diğer Node up durumdaydı.

Failover da ise HV-Node1 down olduğu için quorum diski de diğer node üzerine taşınmış durumda.

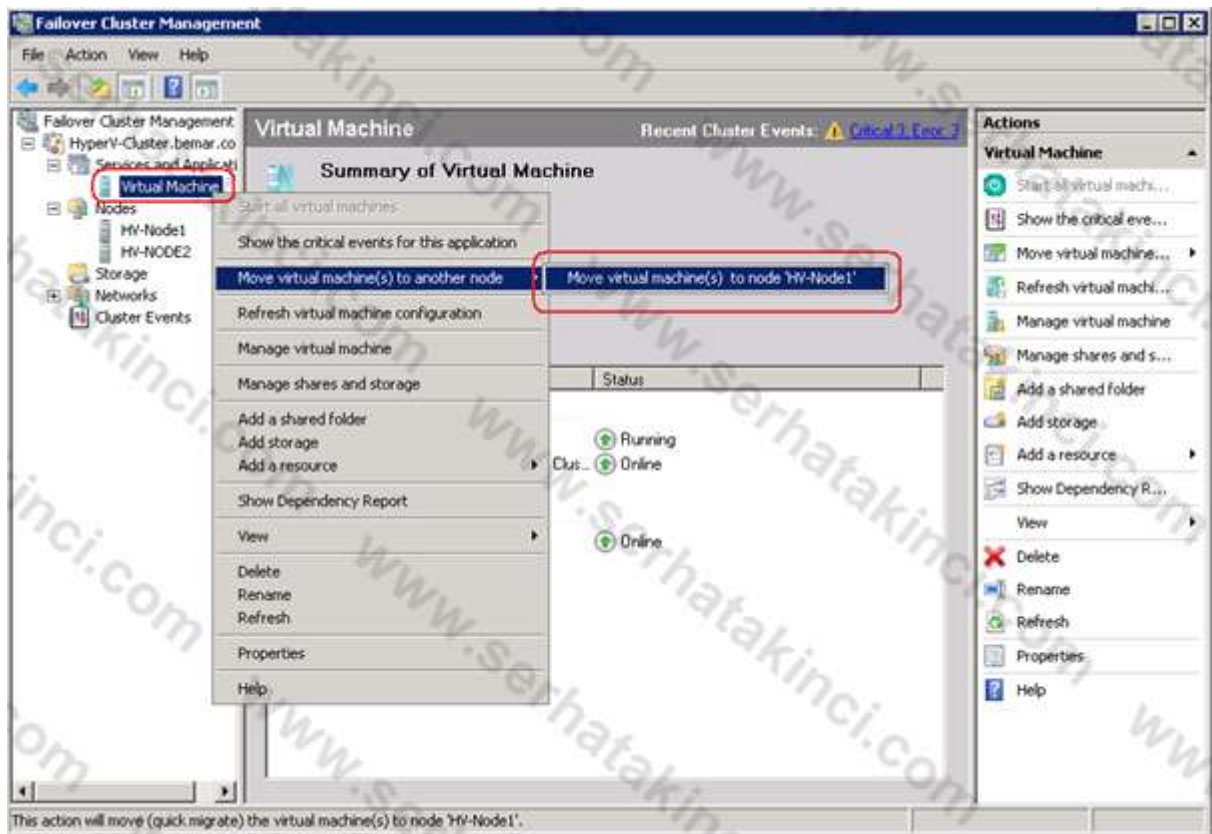


Şimdi down durumda olan HV-Node1'i tekrar açıyorum.

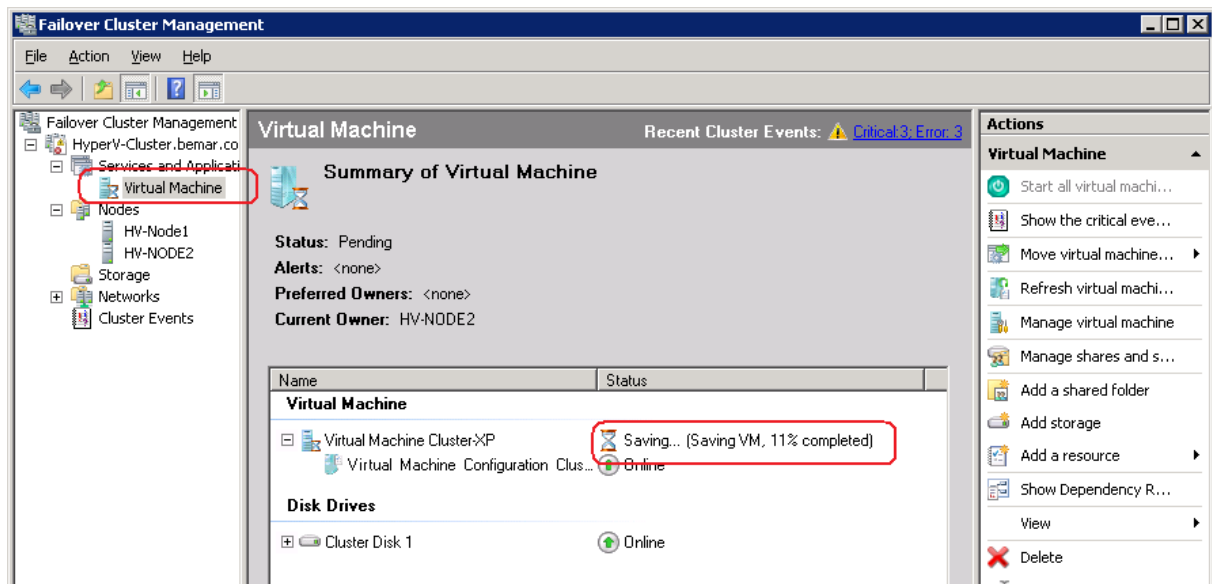
Cluster yapısı hemen HV-Node1'in up olduğunu anlıyor ve bu durumu Failover Cluster Management konsolu üzerinde görebiliyoruz.

Şu an VM'i QM ile eski yerine yani HV-Node1 üzerine alabiliriz yada bu şekilde bırakabiliriz.

HV-Node1 üzerine almak için **Move virtual machine(s) to node 'HV-Node1'** dememiz yeterli.

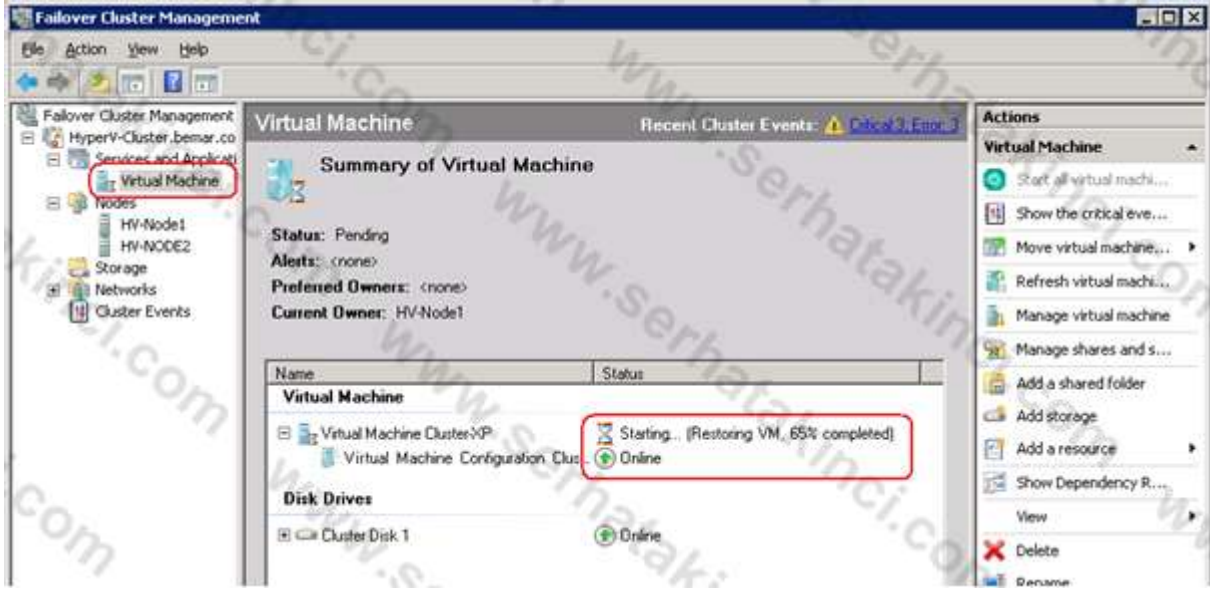


Saving ...

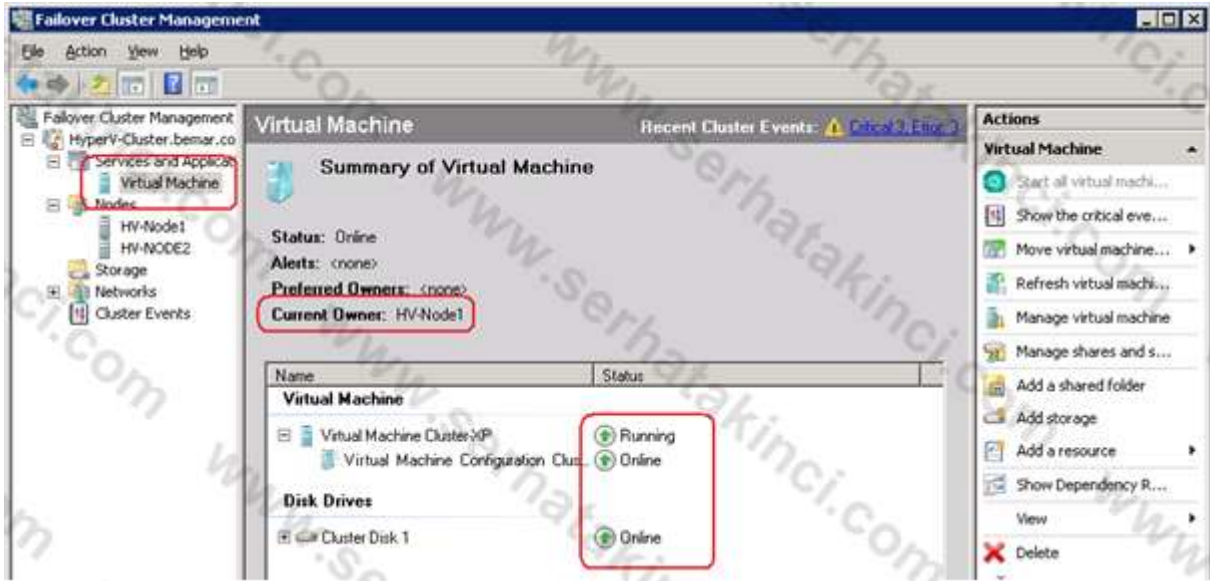


Ve Starting...





Sonrasında **Virtual Machine** servisinin sahibi tekrar **HV-Node1** olur.



Quorum disk ise hala HV-Node2 üzerindedir çünkü QM ile quorum yada diğer clustered disklerin taşınmadığını söylemiştik.

**Quick Migration** ve **Failover(HA)** özelliklerini test ettikten sonra **iSCSI Storage** ile **Hyper-V Failover Cluster** makalemizin sonuna gelmiş olduk.

Bu makalede öğrendikleriniz ile fiber channel storage kullanarak ta aynı yapıyı kurmanız mümkün. Belki bazı konfigürasyon adımlarında değişiklikler olabilir ancak olayın mantığı tamamen aynı.

Bu yapı üzerinde Hyper-V R2 ile gelen **Live Migration** özelliğini de kullanabilirsiniz. Ekstra bir şeye gerek yok. QM yapmış gibi LM yapabilirsiniz. Sadece Host sistemlerin **Windows Server 2008 R2** yada **Hyper-V Server 2008 R2** olması gerekiyor.



## 11. Diğer Kaynaklar

[Hyper-V Kurulumu](#)

[Sanal Makine Yaratılması](#)

[Sanal Makine Kurulumu](#)

[Diğer Failover Cluster Senaryoları](#)

[Diğer Türkçe Makaleler](#)

Konu ile ilgili sorularınız için [serhatakinci@gmail.com](mailto:serhatakinci@gmail.com) adresinden bana ulaşabilirsiniz.