TÜRK HAVA KURUMU ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

YAZILIM TANIMLI VERİ MERKEZİ VE

FELAKET KURTARMA MERKEZİ TEMELLERİ VE MODELLEMESİ

LİSANSÜSTÜ TEZİ

Ali Onur MALKOÇ

BİLİŞİM TEKNOLOJİLERİ

YÜKSEK LİSANS PROGRAMI

ŞUBAT 2018

Abstract

Bir veri merkezi, ağ bileşenleri, sunucu donanımları ve veri depolama sistemlerinden oluşan karmaşık bir ortamdır. Geleneksel olarak bu sistemler sistem yöneticileri ya da alt yapı mühendisleri tarafından ya elle manuel olarak ya da betik yığın programcıkları ile yönetilirler. Bulut bilişim bu geleneksel modeli daha otomasyon tabanlı çözümlere doğru itmiştir. Sunucuların ve hizmetlerin mümkün olduğu kadar hızlı olarak hatta dakikalar içerisinde hazır olmaları beklenmektedir.

Bu tez bir veri merkezinin nasıl hızlı, güvenilir ve ölçeklenebilir otomasyona elverişli halde tasarlanabileceğini incelemektedir. Bunların yanı sıra veri merkezi sanal sunucu göçü gibi yüksek erişilebilirlik ve ölçeklenebilirlik yeteneklerini de sağlamalıdır. Çoklu üretici, tedarikçi ortamı beklenmekle birlikte tüm bileşenlerin merkezi yönetim olanağını desteklemesi gerekmektedir. Alt yapı veri merkezinin tüm bileşenlerinde farklı API’ları (Application Programming Interface) sağlamalıdır. Bu API’ları kullanarak yönetilebilen veri merkezlerine yazılım tabanlı veri merkezi denir.

Yazılım tabanlı veri merkezine dair bir uygulama sunulmaktadır. Veri merkezi VMware sanallaştırma yazılımları, DELL-EMC sunucu donanımları ve veri saklama cihazları ve Cisco ağ ekipmanları ile tasarlanmıştır. Sanallaştırılmış ortam, toplu kurulum ve yapılandırma özellikleri sağlayan yazılımlar sayesinde oluşturulmuştur. Esnek otomasyon iş akışları oluşturmaya elverişli bir orkestrasyon yazılımı kurulmuştur. İhtiyaca uygun düzenlenebilen sanal sunucu oluşturabilen, DNS konfigürasyonu yapabilen, veri depolama tahsis edebilen ve ağ konfigürasyonu yapabilen bir orkestrasyon iş akışı tasarlanmıştır.

Tasarımı iyileştirme adına ek fikirler sunulmuştur. Bu başlıklar ortam güncellenmesi ve güvenlik yamalarının yapılmasına, IPv6 konuşlanmasına ve otomatik yapılandırma yönetimine dairdir. Yapılandırma yönetimi işletim sistemi yapılandırma otomasyonuna izin verdiği için alt yapı yönetimindeki bir sonraki adımdır. Yapılandırma otomasyonu, sunucu oluşturulması gibi alt yapı otomasyonuna dayandığı için bu tezde sunulmaktadır.

Bu sunulan uygulama kanıtlamaktadır ki çoklu üretici/tedarikçi donanımları ile yazılım tabanlı bir veri merkezi kurulması mümkündür. Bununla birlikte tüm bileşenler bir şekilde ya otomasyonu desteklemeli ya da bir API sağlamalıdır. Bu tez içerisinde sunulan uygulama farklı üreticiler tarafından sağlanabilecek farklı bileşenler ile de (donanım ve sanallaştırma katmanı da dâhil) oluşturulabilir.

Önsöz

Tez kapsamında tarif ettiğimiz yöntem bilişim operasyonlarının gerçekleştirilmesi açısından yepyeni bir yoldur ve sistem mimarı olarak gerçekleştirmekte olduğum işim ile yakından ilintilidir. Tasarımın yapılması, ilgili dokümanların yazılması ve tasarımın hayata geçirilmesi HAVELSAN A.Ş. için çalıştığım süre içerisinde Yüksek Seçim Kurulu yeni veri merkezi ve felaket kurtarma merkezi projesi kapsamında gerçekleştirilmiştir.

Tez danışmanım Meltem İmamoğlu’na teşekkürlerimi bir borç bilirim.

1. Giriş

Son 20 yılda bilişim endüstrisi bir seri sanallaştırmaya dair inovasyondan geçmiştir. 2006 yılında AMD ve Intel tüm bir sanallaştırmaya müsait, alt tarafta bulunan donanımı tamamen simüle edebilen CPU komut setleri yayınlamıştır.[[1]](#footnote-1) Bu da bir sanallaştırma sunucusu, bir hipervizör üzerinde birden çok heterojen işletim sistemi örneği çalıştırılabilmesine olanak tanımıştır. O zamandan bu yana benzer yenilikler ağ cihazları ve veri depolama sistemleri içinde uygulanmaya başlamıştır.

* 1. Gerekçe

Bünyesinde sunucular ya da servisler barındıran bir veri merkezine barındırma (hosting) ortamı denilebilir. İçerisinde sunucular, ağ bileşenleri ve veri depolama donanımı, etkin bir havalandırma, soğutma ve yedekli güç sistemleri bulunan kompleks bir ortamdır. Enerji kullanımı yüksek donanımlar barındırdığından enerji tüketimi de haliyle fazladır. Dolayısıyla enerji etkin çözümler çok daha fazla popülerdir.

Sanallaştırma sayesinde istek üzerine oluşturulabilen servisler, daha iyi ölçeklenebilirlik, hata toleransı ve yüksek erişilebilirlik kavramları mümkün olmaktadır. Daha yüksek sunucu konsolidasyonu sebebiyle enerji tüketimi da azalmıştır. Son birkaç yıldır bulut bilişim IT endüstrisinden büyük bir iş kolu haline gelmiştir. Bulut bilişim ölçeklenebilir alt yapı, otomasyon ve farklı ihtiyaçlara göre farklı servis modelleri sunmaktadır. Aynı zamanda bulut bilişim maliyetlerin düşmesine de yardımcı olur[[2]](#footnote-2). International Data Corporation (IDC) firmasının yaptığı analizlere göre bulut bilişim 20 yıl içerisinde bilişim endüstrisini, %26 yıllık bulut hizmetlerinin kullanımının artırmasıyla dönüştürecektir[[3]](#footnote-3).

Geleneksel hosting teknolojileri bulut tabanlı servis ve hizmetleri vermek için yeterli değillerdir. Hizmet ve servislerin provizyonlanmalarında otomasyonun uygulanabilmesi için hosting alt yapısında bulunan tüm bileşenlerin hesaba katılması gerekmektedir[[4]](#footnote-4). Eğer ortamda sanallaştırma teknolojileri kullanılıyorsa bu bileşenlerin çoğu zaten bu tip otomasyonlara müsaittir. Buna rağmen ağ bileşenleri gibi kalıt donanımlar tamamıyla provizyonlama yapılmasını zorlu kılmaktadır. Bununlar birlikte herhangi bir otomasyonu desteklemeyen donanımlar bir sürecin el ile yapılması gereken kısımları olduğu anlamına gelmektedir.

Alt yapı otomasyonu çevik (agile) yazılım geliştirilmesinde de önemli bir role sahiptir. Çevik yazılım geliştirilmesi hızlı geri dönüş yapılmasını gerektirir. Bu da yazılımın derlenmesi, konuşlandırılması ve test edilmesi süreçlerinin kesintisiz bir süreç olarak gerçekleşmesi anlamına gelmektedir. Bu süreci tarif eden popüler adlardan birisi sürekli teslimattır (Continuous Delivery) ve otomatikleştirilmiş provizyonlama da böyle bir sürecin temel gereksinimidir. Bu IT operasyonları için bir zorluk teşkil etmektedir. IT operasyonları ile yazılım geliştiricilerin kesintisiz olarak birbirleri ile etkileşimlerine DevOps (Development and Operations) adı verilmektedir.[[5]](#footnote-5)

Bahsettiğimiz otomasyona bir örnek vermek gerekirse bu da yeni bir sunucunun provizyonlanması sürecidir. Yeni bir sunucu oluşturulduğunda ona CPU ve RAM gibi kaynaklar atanır. Bu kaynaklar bir sanallaştırma sunucusundan atanır. Kurulumdan sonra sunucu için otomatik olarak bir ağ yapılandırması yapılması ve kullanacağı depolama alanının belirlenmesi gerekmektedir. Bütün bu sürecin hızlı bir şekilde tamamlanması gerekmektedir. Sürekli teslimat (Continuous Delivery) kapsamında sunucunun üzerinde bir yazılım ortamının kurulabilmesi mümkün olmalıdır ki bu da ancak alt yapı otomasyonu sayesinde gerçekleşebilir.

Böyle bir otomasyon sistemi oluşturulabilmesi için, tüm hosting ortamının donanım katmanından uygulama katmanına kadar otomasyonu destekleyecek şekilde tasarlanmış ve kurulmuş olması gerekmektedir. Sanallaştırma, ağ, veri depolama ve yönetim bu sürecin ana yönlerindendir. Mevcut veri merkezinin yazılım tabanlı veri merkezine güncellenmesi yerine yeni bir hosting ortamının tasarlanıp oluşturulması gerekmektedir. Mimari açıdan tasarımın ayrık ve net temel ilkeleri içermesi gerekmektedir. Bunun sayesinde benzer ortamların aynı özellik seti ile kurulması mümkün olabilecektir. Ama aynı mimari adapte edilerek farklı donanımlar ile tekrar kullanılabilecektir.

Yazılım tanımlı ya da yazılım tabanlı veri merkezi terimi VMware firması tarafından bir veri merkezi için böyle bir mimariyi tarif etmek için icat edilmiştir. Bilişsel, ağ ve veri depolama sanallaştırılmasına dair alınması gereken mimari kararları barındırır.[[6]](#footnote-6) Ağ sanallaştırılmasına yazılım tabanlı/tanımlı ağ (Software defined networking, SDN) ve veri depolama sanallaştırılmasına da benzer bir şekilde yazılım tabanlı/tanımlı depolama (software defined storage, SDS) denmektedir.

* 1. Hedefler

Bu tezin amacı VMware sanallaştırılması kullanılarak nasıl bir yazılım tabanlı veri merkezi tasarlanıp oluşturula bilineceğini araştırmaktır. Her ne kadar burada tarif edilen yöntemler VMware sanallaştırma alt yapısına özgü olsa da diğer sanallaştırma teknolojileri de değerlendirilmiş ve aynı tip özellikleri sağladıkları gözlenmiştir.

* 1. Tezin Yapısı

Yazılım tabanlı ver merkezi oluşturulmasına dair teori 2. bölümde sunulmaktadır. Yazılım tabanlı veri merkezi ile sıradan barındırma teknolojilerinin kıyaslanması da bu bölüm içerisinde sunulmaktadır. 3. Bölümde yazılım tabanlı bir veri merkezi uygulaması anlatılmaktadır. Bu uygulama içerisinde bahsi geçen yazılım tabanlı veri merkezi için gerçekleştirilen mimari tasarım kararları, otomasyon uygulaması ve kaynak yönetimi de bulunmaktadır. Ek durumlar ve gelecekte yapılması beklenen geliştirmeler 4. bölümde anlatılmaktadır ki burada henüz uygulamaya geçirilmemiş durumlardan ve problemlerden bahsedilmektedir. Son olarak da 5. bölümde sonuç ve çıkarımlar sunulmuştur.

1. Yazılım Tanımlı Veri Merkezi

Bu bölümde yazılım tanımlı veri merkezi teorisi verilmektedir. Anahtar bileşenler ve teknolojiler anlatılarak gözden geçirilmektedir. Öncelikle sıradan barındırma hizmetlerinde yaşanan problemler ve sınırlamalar sunularak tartışılmaktadır. Akabinde sanallaştırma platformları incelenerek sunulmaktadır. VMware sanallaştırma platformu ve yazılım tanımlı veri merkezi kabiliyetleri sunularak bir mimari örneği tanıtılmaktadır. Beraberinde SDDC’ye veri saklama kabiliyetleri sunan bir yazılım tabanlı saklama teknolojisi sunulmaktadır. Son olarak da ağ sanallaştırılması çözümleri ve çeşitli güvenlik standartları değerlendirilmektedir.

* 1. İçeriğin Gözden Geçirilmesi

Yazılım tanımlı veri merkezi, bir veri merkezi tasarlanırken kullanılan mimari bir modeldir. İşlem yapabilme, veri saklama, ağ ve güvenlik kaynaklarını yalıtma kabiliyetine haizdir. Öyle ki bu sanallaştırılmış kaynakların oldukça yüksek şekilde otomasyon ile konuşlandırılmasını ve yönetilmesini sağlar.[[7]](#footnote-7) Bu da IaaS (Infrastructure as a Service, Alt yapı hizmeti), PaaS (Platform as a Service, Platform hizmeti), SaaS (Software as a Service, Yazılım hizmeti) gibi çeşitli bulut bilişim modellerinin kullanımını mümkün kılar. Genel olarak geleneksel ve daha evvel kullanılan mimari ve teknolojiler ile erişilemeyecek bir IT etkinliği ve çevikliği sağlar.[[8]](#footnote-8)

SDDC, ITaaS (It as a Service, IT hizmetleri) denen operasyonel bir model ile yakından ilişkilidir. Bu modelde IT organizasyonu bulut hizmetlerini (hem harici hem de dahili) self servis portaller aracılığıyla sunarak ayrı bir kurum ya da iş yeri gibi çalışır. Bu modeli mümkün kılan alt yapı ancak yazılım tanımlı bir veri merkezinde olanaklıdır.[[9]](#footnote-9)

Sanallaştırma hem sunucu hem de veri saklama sistemleri için tasarım kararları vermek suretiyle çeşitli teknolojiler ile sağlanabilir. Bu teknolojiler ikiye ayrılırlar; native sanallaştırma (?) ve hosted sanallaştırma (?).

Native sanallaştırmada sanallaştırmayı sağlayan hipervizör ana işletim sistemi olarak donanım katmanının üzerinde çalışmaktadır. Tüm sanal makineler hipervizör katmanının üzerinde çalışırlar. Hosted sanallaştırmada ise hipervizörü çalıştırabilmek için üzerinde çalışacağı bir temel işletim sistemine ihtiyaç duyulmaktadır. Sanal makineler ise bu temel işletim sisteminin üzerinde çalışan hipervizörün üzerinde bir katmanda çalışırlarki bu da fazladan bir overhead(?) e yol açar.

Kurumsal bir ortamda native sanallaştırma bir standartlaşmış durumdadır. Daha yüksek verimlilik, erişilebilirlik ve fazladan bir yazılım katmanı daha olmadığından yüksek bir güvenlik seviyesi sağlarlar.

* 1. Sıradan Barındırma Hizmetlerindeki Sınırlamalar, Zorluklar

Sanallaştırma veya bulut bilişimden önce hosting ya paylaşılan ya da adanmış (?) sunucular üzerinden sağlanmakta idi. Sunucu konumlandırılıp yapılandırılmaları el ile yapılmaktaydı ki bu da zaman kaybına yol açmaktaydı. Ayrı sunucular üzerinde çalışmaya ihtiyacı olan uygulamalar için kendilerine tahsis edilmiş sunucu fiziksel sunucular olması gerekli idi. Bir veri merkezinde bu da sunucu donanımları için yüksek maliyet, fiziksel alan ve elektrik ihtiyacı demekti[[10]](#footnote-10). Aynı zamanda bu yapı tek bir donanımdan kaynaklanabilecek arızalarda tüm veri merkezinin etkilenmesi söz konusu olabilmekteydi.

Sanallaştırma bu problemleri basitleştirdi. Sanallaştırma sayesinde sunucuları başka sunucular üzerinde konsolide edilebilmeye başlandı. Tahsis edilmiş fiziksel sunucuların düşük kullanımları bir sorun iken tek bir fiziksel sunucu üzerinde birbirlerinden yalıtılmış birden fazla sanal sunucu konuşlandırılmaya başlandı. Bu da toplamda daha az fiziksel sunucu, dolayısıyla daha az fiziksel yer ihtiyacı ve elektrik demektir. Sonuç olarak donanım maliyetleri de azalmış oldu. [[11]](#footnote-11)

Bunlara ek olarak daha önce sağlanması oldukça güç olan çeşitli mekanizmalar sanallaştırma sayesinde mümkün oldu. Yüksek erişilebilirlik, ölçeklendirilebilme ve felaket kurtarma gibi özellikler sanallaştırma sayesinde kolayca erişilebilir hale geldi. Paylaşımlı veri saklama sistemleri kullanılarak ve sanallaştırma sunucuları küme haline getirilerek birisinde gerçekleşebilecek bir arızada diğer küme elemanları arızalananın görevini devralarak hizmet kesintisinin önüne geçebilmektedir.

* 1. Sanallaştırma Platformları

Sunucu sanallaştırması pek çok değişik teknoloji ile sağlanabilir. Dolayısıyla bir yazılım tabanlı veri merkezi VMware’in vSphere yazılımı dışındaki diğer sanallaştırma platformları ile de sağlanabilmektedir. Citrix’in XenServerı veya Microsoft’un Hyper-V siz gibi sanallaştırma platformları da bir yazılım tabanlı veri merkezi için elzem olan benzer fonksiyonlara sahiptirler. Örnek vermek gerekirse orkestrasyon yazılımlarına olan destek ve canlı sanal makine göçü gibi özellikler gereklidir. Bütün bu ortamlar arasında alt yapıyı oluşturan bileşenler hemen hemen aynıdır.

Bu tezimiz içerisinde bir yazılım tabanlı veri merkezi sunulmakta ve VMware ürünleri ve teknolojileri kullanılarak tasarlanmaktadır. Bu teknolojilerin seçilmesindeki sebep uçtan uca tüm bir yazılım tabanlı veri merkezi için gerekli olan tüm yetenekleri barındırması, son derece yaygın olarak bu ürünlerin kullanılması, tüm ürün ve teknolojilerin aynı eko sistem içerisinde birbirleri ile etkin bir şekilde entegre çalışabilmesi ve son olarak da tüm ürünlerin ve bu ürünlerin bir SDDC için nasıl kullanılması gerektiğine dair oldukça çok dokümantasyona sahip olmalarıdır.

* 1. Mimari ve Bileşenleri

Sanallaştırma platformu veri merkezlerini, sunucu kümelerini ve sanallaştırma sunucularını içermektedir. VMware üzerinde bu platforma vSphere adı verilmektedir. Fiziksel anlamda bakılırsa bir veri merkezi sunucu donanımlarını, veri depolama sistemlerini ve ağ ekipmanlarını içerisinde barındıran bir bütün yapıdır. Bir SDDC açısından bakıldığında ise bir ya da birden fazla sanallaştırma sunucularının birleşerek oluşturduğu sanallaştırılmış küme sunucular demektir. Bir küme yüksek erişilebilirlik ve ölçeklenebilirlik sağlayan transparan bir grup sunucudur. Aynı zamanda buna bir kaynak havuzu da denebilir.

Yukarıda tanımlandığı gibi bir küme birden çok sanallaştırma sunucularının birleşiminden oluşabilir. Bir küme üyesi iken sanallaştırma sunucuları kendi aralarında küme içerisindeki sanal makineler hakkında bilgileri paylaşırlar. Bir sanal makine arızası ya da bütün bir sanallaştırma sunucusunun arızası durumunda bu durumdan etkilenen tüm sanal makineler faal olan bir başka sanallaştırma sunucusuna aktarılırlar.

Bir sanallaştırılmış VMware alt yapısı ya da vSphere ortamı aşağıdaki bileşenlerden oluşur:

* Hipervizör denilen yalın metal sanallaştırma sunucuları. Hipervizör hafıza üzerinde çalıştığı ve sanal makineler paylaşımlı başka veri depolama sistemlinde bulundukları için minimal veri depolama alanına ihtiyaç duyarlar. VMware’de bunlara ESXi host adı verilmektedir.
* Tüm verilerin saklandığı ve korunduğu bir veri depolama sistemi. Bir kümeye üye olan tüm sanallaştırma sunucularının aynı veri saklama mekanizmalarına erişimleri olması gerekmektedir. Bu veri saklama mekanizmalarına datastore denilmektedir ve tüm ESXi hostlar bunlara bağlantılıdır.
* Yönetim birimi ya da birimleri ki alt yapıyı kontrol ve yapılandırmada kullanılırlar. VMware de buna vCenter sunucusu denilmektedir.

Virtualization platform itself is not dependent on the virtualization host hardware. VMware ESXi runs on a range of different vendors and CPU architectures. Compatibility between the hypervisor software and server hardware can be determined. However, to prevent compatibility problems between hosts, it is advised to use homogenous server hardware within a cluster.

Sanallaştırma platformunun kendisi sanallaştırma sunucu donanımına bağımlı değildir. VMware ESXi farklı üreticilerin farklı işlemci mimarileri üzerinde çalışabilmektedir. Hipervizör yazılımı ile sunucu donanımı arasındaki uyumluluk belirlenebilmektedir. Ancak sanallaştırma sunucuları veya küme üyeleri ve hatta birden fazla küme arasındaki uyumluluğun sağlanabilmesi ve problemlerin azaltılabilmesi adına homojen bir sunucu donanımı seçilmesi tavsiye edilmektedir.

* 1. Sanallaştırma Platformu Yönetimi

VMware vCenter sunucusu VMware sanal alt yapısını merkezi olarak yönetmek için kullanılmaktadır. Vcenter kendisi için özel olarak ayrılmış bir donanım üzerine kurulabileceği gibi bir sanal makine olarak kendi yönettiği sanal alt yapı içerisinde de bulunabilir. Birden çok veri merkezini, küme sunucuları ve sanallaştırma sunucularını yönetebilmektedir. Yönetim yetenekleri yanı sıra veri merkezi çerçevesinde takip ve izleme yapabilmektedir.[[12]](#footnote-12)

* + 1. Sanal Makine Provizyonlama

Sanal makineler vCenter aracılığıyla provizyonlanırlar. vCenter alt yapı içerisindeki tüm sanallaştırma sunucularına bağlı olduğu için, sanal makineler belirli bir küme üyesi belirli bir sanallaştırma sunucusuna atanabilirler. Bir sanal makine, bir veri depolama sistemi üzerindeki bir veri birimi ile ilişkilendirilmiş bir datastore üzerinde yer alırlar. Datastore üzerinde bu sanal makine için bir klasör oluşturulur ve bu klasör içerisinde sanal makine ile ilgili yapılandırma dosyaları ile sanal makinenin sanal diski yer alır. Provizyonlanma esnasında birden fazla diskin eklenmesi ve bu disklerin boylarının belirlenmesi mümkündür. Ağ kartı gibi her türlü uç birimin sanal makineye eklenerek tanımlanabilir. Bütün bu yapılandırmalar klasör içerisindeki bir dosyaya yazılarak kalıcı hale getirilir. Bu yapılandırma içerisinde işlemci ve hafıza miktarları da bulunmaktadır.

Genellikle sanal makineler içerisinde bir daha önceden kurulmuş bir işletim sistemi olmadan provizyonlanmazlar. Yoksa el ile bir işletim sistemi kurulumu yapılması gerekir. Her ne kadar kurulum adımları da çeşitli teknikler ile otomatize edilebilse de daha hızlı olan yöntem daha önceden işletim sistemi kurulmuş olan bir sanal makine şablonundan yeni bir sanal makine üretilmesi ve bu yeni sanal makinenin kendisine özgün olarak yapılandırılmasıdır. Bu şablon sanal makineler sadece önceden kurulmuş işletim sistemlerini barındıracak kadar disklere ve çevre birimlerine sahiptirler. Sıfırdan yeni bir sanal makine oluşturulup işletim sistemi kurulmasındansa şablondan üretilen sanal makineye sadece işletim sistemi özelinde ayar yapılması yeterli olacaktır.

Sanal makine oluşturulmasının üçüncü bir yolu ise klonlama yöntemidir. Şablondan bir sanal makine üretilmesine oldukça benzemektedir. Üzerinde işletim sistemi ve yapılandırması bulunan mevcut bir sanal makinenin aynısından bir tane daha sanal makine üretmek mümkündür. Birbirlerine benzeyen sanal makineler oluşturulması için uygun bir yöntemdir. Sadece sanal makinelerin temel yapılandırmaları değişiklik göstermektedir.

* + 1. Sanal Makine Göçü

Sanal makineler bir sanallaştırma sunucusundan bir başkasına, VMware’in vMotion adı verilen bir yeteneği sayesinde canlı olarak göz ettirilebilirler. Bu özellik kaynakların eşit olarak bir küme içerisinde ya da bir veri merkezi içerisinde etkin olarak dağıtılabilmesi için faydalıdır. Hem hedef hem de kaynak sanallaştırma sunucularının aynı kaynaklara erişebiliyor olması gereklidir. Bu göç işlemi sanal makinenin kullandığı fiziksel hafızanın kopyalanması ile başlar. Bu üç adımda gerçekleşir:[[13]](#footnote-13)

* Guest trace fazı: Göç ettirilecek misafir işletim sisteminin hafıza sayfalarının işaretlenmesi ile başlayan fazdır. Sanal makine çalışır vaziyette olduğundan hafıza sayfaları devamlı değişim içerisinde bulunduğundan bu sayfaların işaretlenmesi gereklidir.
* Kopyalama öncesi fazı: Hafıza sayfaları iterasyonlar halinde kopyalanmaya başlanır. Öncelikle tüm hafıza haritası kopyalanır, sonra da bir önceki iterasyondan itibaren değişen sayfalar kopyalanır.
* Değişim fazı: Sanal makine bir an için sersemletilerek tüm I/O işlemleri dondurulur. Bu sayede son iterasyondan beri değişen hafıza sayfaları kopyalanırken yeni bir değişim olmasına engel olunur. Kopyalama sonrasında orijinal sanallaştırma sunucusundan bu sanal makineye atanan kaynaklar düşürülür ve göç ettirildiği yeni sanallaştırma sunucusu üzerinde yeniden bu kaynaklar atanır.

Sanal makinenin işlemci, ağ ve diğer sanal cihazlarının son halleri de kopyalanır. Tüm aktif ağ bağlantıları göç esnasında korunur. Kopyalama tamamlandığında sanal makinenin işlemlerine devam etmesine izin verilerek hedef sanal sunucu üzerinde kesintisiz olarak çalışması sağlanır.

Eğer sanal makinenin bir kümeden başka bir kümeye ya da bir veri merkezinden başka bir veri merkezine göç ettirilmesi gerekirse, vSphere 5.1 öncesindeki sürümlerde sanal makinenin kapatılması gerekli hale gelir. Paylaşılan ortak veri depolama alanları küme sunucular ya da veri merkezlerinin aynı anda erişilebilir olmaması halinde bu durum geçerlidir. Canlı göç yapılmasına kıyasla bu çok daha basit bir operasyondur. Sanal makine kapalı durumda olduğundan sadece veri depolama alanında saklanan verilerin kopyalanması gereklidir. vSphere 5.1’den sonraki sürümlerde gelen Storage vMotion özelliği ile birlikte bir sanal makinenin kümeler, veri merkezleri ve hatta vCenterlar arasında kesintisiz olarak göç ettirilebilmesi mümkün olmuştur. Storage vMotion kendi başına sanallaştırma sunucu değiştirmeden sanal makinenin üzerinde barındığı datastore, dolayısıyla veri depolama cihazı kesintisiz olarak değiştirilebilmektedir. Hem vMotion hem de Storage vMotion özelliği bir arada sırayla kullanılarak daha önceden hiçbir ortak veri depolama alanı bulunmayan iki sanallaştırma sunucusu, küme sunucular ve veri merkezleri arasında da kesintisiz olarak göç yapabilmek mümkündür. Farklı veri depolama alanları arasında göç yaptırılarak sanal makinenin çeşitli servis seviyelerinde çalışması mümkündür.

* + 1. Kaynak Havuzları

Kaynak havuzları hafıza ve işlemci gibi kaynakların sanal makineler tarafından nasıl kullanılacaklarını yönettikleri yapılardır. Buna soyut kaynak bölümlendirme denilmektedir. Kaynak havuzları bir küme içerisinde ya da tek bir sanallaştırma sunucusu üzerinde de tanımlanabilmektedir. Kaynak havuzları oluşturarak bir grup sanal makinenin mümkün olduğunca kendi aralarında paylaştıkları kaynakları belirleyip sınırlandırmak mümkündür. Bunun sayesinde değişik servis seviyeleri oluşturabilir. Belirli bir sanal makineye kullanması için belirli kaynaklar adanabilir ve aynı sanallaştırma sunucusu üzerindeki bir başka sanal makine o kaynakları kullanamaz. İlk sanal sunucu kendisine adanmış olan kaynakları ister ihtiyacı olsun ister olmasın sanallaştırma sunucusu tarafından bu kaynaklar her koşulda sanal makineye atanır. Diğer sanal makinelerde kaynak ihtiyacı olsa da adanmış olan kaynaklar düşüldükten sonra geriye kalan kaynaklar arasından hisselerine göre kullanırlar.

Kaynak havuzlarının bir başka özelliği de homojen olmayan veri merkezleri içerisinde ya da birbirlerinden farklı miktarlarda fiziksel kaynağa sahip küme sunucular arasında performans tutarlılığını sağlamaktır. Bir sanallaştırma sunucusu ya da küme sunucular üzerinde belirli oranlar ile çalışan bir sanal makine aynı oran ile başka bir sanallaştırma sunucusu ya da küme sunucular üzerine taşındığında aynı işlemci hızlarına sahip olmayan sunucular yüzünden aynı oran ile daha çok ya da daha az kaynağa sahip olabilirler. Kaynak havuzu tüm havuz için belirli miktarda kaynağı somut olarak ayırır ve bir sanal makine de kaynaklara erişiminin oranına bu dahil olduğu havuzun tümü üstünden sahip olur.

* + 1. Yüksek Erişilebilirlik

Küme yüksek erişilebilirliği VMware’in High-Availability modülü sayesinde sağlanır. Bu özellik vSphere sanallaştırma platformu içerisinde, Hipervizör seviyesi yüksek erişilebilir ajanları ile ve hatta sanal makine işletim sistemleri seviyesinde sağlanabilmektedir.[[14]](#footnote-14) Uygulama sürekli takip edilen bir heartbeat takibi ile yapılmaktadır. Bir sanallaştırma sunucusu arızası durumunda diğer sanallaştırma sunucuları arızalanan sunucudan belirli bir süre boyunca haber alamazlarsa o sunucu üzerinde çalışan sanal makinelerin taşınması gerektiğini fark ederler. Bu operasyona failover denilmektedir. Eğer bir sanal makine işletim sistemi içerisindeki ajanlar sanallaştırma sunucusu üzerindeki HA ajanları ile belirli bir süre boyunca iletişime geçemezlerse HA ajanları işletim sisteminin yeniden başlatılmasına karar verebilir. Tüm bu yetenekler büyük ölçüde paylaşımlı veri depolama sistemlerine dayanırlar. Bir küme içerisindeki üye sanallaştırma sunucuları HA ajanları birbirleri ile hem yönetim ağı üzerinden hem de veri depolama sistemi üzerinden iletişime geçmeye çalışırlar. Ağ izolasyonu gibi durumlarda bir sanallaştırma sunucusu üzerindeki sanal makineleri gereksiz yere yeniden diğer üyeler üzerinde başlatmadan önce, ağ üzerinden erişilemeyen sanallaştırma sunucusunun sanal makinelerinin veri depolama üzerinde hala kilitli durumda olup olmadıkları kontrol edilir. Bu durumda nasıl davranacakları kullanıcı tarafından belirlenebilir. Bir failover durumunda sanal makinelerin diğer sanallaştırma sunucuları üzerinde yeniden çalıştırılabilmesi için gerekli kaynakların müsait olduğu sürekli kontrol edilir.

* + 1. Ölçeklenebilirlik

Sunucu kümeleri, daha çok sanallaştırma sunucuları eklenmek suretiyle yatay olarak ölçeklenebilir. VMware HA bir küme içerisinde 32 adede kadar sanallaştırma sunucusu barındırabilir. VMware HA, Dağıtık Kaynak Planlama (Distributed Resource Scheduler, DRS) özelliği ile birlikte kullanılırsa tüm kaynaklar küme içerisinde etkin bir şekilde yayılabilir. Gerektiğinde kaynakların dengeli bir şekilde dağıtılması adına sanal makineler bir sanallaştırma sunucusundan diğerine otomatik olarak göç ettirilebilir. Ölçeklendirme bir kümenin bakım operasyonları esnasında büyük önem kazanır. Tipik olarak bir sanallaştırma sunucusu donanımın değiştirilmesi, tamiri ya da yazılım güncellenmesi veya yükseltilmesi esnasında yeniden başlatılabilir. Bu sanallaştırma sunucusu üzerindeki tüm sanal makinelerin diğer bir başka sunucuya göç ettirilmesi gerekir ki bu da ancak kümenin kaynak kapasitesi aşılmadığında mümkündür. Bir küme içerisinde en az bir sanallaştırma sunucusunun tamamıyla boşaltılabilmesi gereklidir. DRS ve HA seçenekleri arasında bir kümenin arıza durumunda sanal makinelerin yeniden kalan sunucular üzerinde başlatılabilmesi için ne kadar kaynağı âtıl olarak tutması gerektiği belirlenebilir. Bu miktar ister kaç adet sanallaştırma sunucusunun arızası olarak somut bir miktar, ister yüzde olarak toplam küme kaynaklarının bir birimi cinsinden soyut olarak tanımlanabilir.[[15]](#footnote-15)

* + 1. Orkestrasyon Yazılımları ve Otomasyon

Otomasyon, bir yazılım tabanlı veri merkezinin kilit yönlerinden birisidir. Bulut bilişim ve hızlı servis provizyonlama gibi istenildiği anda oluşturulabilen servisler ancak orkestrasyon uygulanarak sağlanabilir. Kaltz, C. orkestrasyonu “dahili ve harici web servisleri ile etkileşebilen yürütülebilir iş süreci” [[16]](#footnote-16)olarak tanımlamaktadır. Asıl orkestrasyon, orkestrasyon motoru tarafından işletilir. Orkestrasyon motoru gerçek iş süreçlerini tarif eden iş akışlarını yürütür. Bir orkestrasyon iş akışı işlem, veri depolama ve ağ katmanlarında bulunan API’ları kullanır. Bu genişletilebilir ve özelleştirilebilir olmalıdır ki daha da fazla sistemler iş akışı sürecine dahil edilebilsin.

Orkestrasyon VMware otomasyon ürünleri ile ya da vCenter’ın uygulama programlama arayüzleri (API) sayesinde gerçekleştirilebilir. Bu API’lar pek çok yazılım geliştirme kitleri (SDK) tarafından erişilebilir ve otomasyon ürünleri tarafından değerlendirilebilir. Sistem yöneticileri vCenter yönetim yazılımında önceden tanımlı iş akışlarını ek bir yapılandırma yapmaya gerek olmadan çalıştırabilirler. Önceden tanımlı iş akışlarına örnek vermek gerekirse bir sanal makinenin klonlanması verilebilir. Bu iş akışında mevcut olan bir sanal makineden tamamıyla aynı bir başka sanal makine yaratılmasıdır. Özel iş akışları yine vCenter API’nı değerlendiren özel yazılımlar ya da betik dosyaları ile oluşturulabilirler. Bununla birlikte esnek ve özelleştirilebilir otomasyon sağlayabilmek için vCenter Orchestrator da (vCO) özel iş akışları oluşturmak için kullanılabilir.

VMware vCenter Orchestrator java tabanlı, bağımsız ve vCenter’a ilişkilendirilmiş bir web yazılımdır. vCenter tarafından sunulan alt yapıyı kontrol etmek için vCenter API’nı kullanır. Orchestrator farklı alt yapıları ve sanal sunucuları yöneten birden fazla vCenter yazılımlarına bağlanabilir. Aşağıda tek bir Orchestrator yazılımının yine tek bir vCenter sunucusunu kontrol ettiği örnek bir mimari gösterilmektedir. VCenter, Orchestrator tarafında bir API çağırısı aldığında, adanmış bir yönetim ağı üzerinden hedef sanallaştırma sunucusunu ya da sunucularını yönetir. Orchestrator aynı zamanda tüm bir küme sanal sunucuyu yönetebilir ve örnek vermek gerekirse bunların aralarında sanal makine göçü gibi bir iş akışı tetikleyebilir.

Orchestrator özel iş yükleri tanımlama kabiliyeti ve Javascript betik motoru sağlar. Tüm yapılandırma Orchestratora bağlantısı olan java tabanlı bir masa üstü yazılım sayesinde gerçekleştirilir. İş akışları bir grafik ara yüz aracılığıyla tasarlanıp oluşturulur. Bu iş akışlarının ihtiyacı olabilecek tüm parametre girişleri ve verebileceği çıkış değerleri veya sonuçları ayarlanabilir. Bir Orchestrator iş akışı adım adım çalışan aksiyonlardan oluşur. Aksiyonlar temel iterasyonlar ve önceden tanımlı API çağırıları olabileceği gibi tamamıyla JavaScript ile yazılmış özel fonksiyonlar da olabilir. Bu da vCenter tarafından sunulan API’ların kompleks bir şekilde kullanılması olanak sağlar. Hata işleme ise exceptionlar ya da iş akışının döndürebileceği aksiyon kodları sayesinde uygulanabilir.[[17]](#footnote-17)

İş akışları masa üstü istemciden başlatılabilir. Yazılım kontrollü bir otomasyon sağlayabilmek için Orchestrator iş akışı özeli API’lar sağlar. Tüm iş akışlarının kendilerine özgü birer tanımlayıcıları bulunur. Bu da Orchestrator’a bir tanımlayıcı ile API çağırısını hedefleyerek bir HTTP POST isteği göndermek suretiyle iş akışının uzaktan yürütülmesine imkân vermektedir. İş akışı parametre girişleri XML ya da JSON biçiminde olabilmektedir.

* 1. Veri Saklama Sistemleri

Kurumsal bir veri merkezinde veri depolama ve yönetimi önemli bir bileşendir. En yaygın olarak kullanılan veri depolama sistemi üreticileri DELL-EMC, NetApp, HPE ve Hitachi’dir.

Veri depolama olarak kullanılabilecek en basit sistem yerel veri depolama kaynaklarıdır. Sanallaştırma sunucusunun kendi üzerinde barındırdığı SAS ya da SATA ara yüzlere sahip doğrudan bir HBA aracılığıyla erişilir. Yerel veri depolama doğrudan bağlı depolamanın (direct attach storage ya da DAS) bir türevi sayılabilir. Bu yaklaşımın, tüm veri tek bir sanallaştırma sunucusu üzerinde barındığı için çeşitli riskleri bulunmaktadır. Eğer bu sanallaştırma sunucusunda bir donanım arızası gerçekleşirse veriye olan erişim mümkün olamayacağı gibi veri tamamen kaybolabilir de. RAID gibi çeşitli direnç katan teknolojiler kullanarak bir miktar iyileştirme sağlanabilse de yerel depolama diğer seçeneklere kıyasla ölçeklendirmesi sınırlı ve yedeklemesi problemli bir yaklaşımdır.

Yüksek riskli yerel depolama kullanmak yerine, veri depolamada yüksek erişilebilirliğe ulaşmanın en sık rastlanan yöntemi bir veri depolama sistemi kullanmaktır. Bir veri depolama sisteminde ayrık depolama kontrol birimleri ve bu birimlere bağlı bir veya birden fazla disk sistemleri bulunur. Bir disk sistemi üzerinde homojen ya da heterojen yapıda SATA veya SAS ara yüzlere sahip SSD ya da manyetik diskler bulunabilir. Fiziksel diskler bir disk rafı içerisinde barınırlar ve bu raf aracılığıyla kontrol birimine bağlıdırlar. Disk rafları Fibre Channel ya da SAS gibi çeşitli protokoller ile kontrol birimine ve birbirlerine bağlanabilirler. Bir veri depolama sistemi kullanım tarzına göre SAN (storage area network) ya da NAS (network access storage) şeklinde adlandırılabilirler.[[18]](#footnote-18)

* + 1. Kümeleme ve Hata Toleransı
    2. Flash tabanlı disk Cache (?)
    3. Saklama Sistemleri Otomasyonu
  1. Ağ teknolojileri
     1. Anahtar Sanallaştırması
     2. Virtual LAN (VLAN)
     3. Virtual Private Network (VPN)
     4. Güvenlik Duvarları ve Güvenlik
  2. Veri Merkezleri için Güvenlik Standartları

1. Mimariye Genel Bakış

Yazılım tabanlı veri merkezi tasarımı, bir IT organizasyonunun sıradan ve tekrarlanabilir isteklerinin karşılanmasını, otomatikleştirmesini ve iş gereksinimlerine çevik ve tahmin edilebilir bir şekilde yanıt verilmesine olanak sağlar. Geleneksel olarak bu tasarıma IaaS (Infrastructure as a Service) ya da alt yapı hizmeti denmektedir. Bununla beraber yazılım tabanlı veri merkezi tipik bir IaaS çöüzümünü genişleterek daha geniş bir kapsam sunar ve daha bütün bir IT çözümü sağlar.

Bu mimari pek çok katman ve modül üzerine dayanmaktadır. Bu da birbirleri ile değiştirilebilir bileşenlerin SDDC gibi bir nihai çözümün ya da sonucun parçası olmasını mümkün kılar. Eğer belirli bir bileşen tasarımı herhangi bir sebepten dolayı bir iş ya da teknik gereksinime uymuyor ise bileşenin kendisine benzer bir başkası ile değiştirilmesi mümkün olmalıdır. Burada sunduğumuz mimari önerisi ve benzer mimarilerin tasarlanması için takip edilmesi gereken adım ve alınması gereken karar silsilesi bu işi gerçekleştirmenin yollarından bir tanesidir. Hem bileşenler hem de nihai mimari dayanıklılık, ölçeklendirilebilirlik uyumluluk adına sahada test edilmiştir. Sonuç olarak sistem istenen IT hedefine ulaşılması için tasarlanmıştır.

* 1. Fiziksel Katman

Çözümün en alt katmanı bazen çekirdek katmanı olarak da anılan fiziksel katmandır. Bu katman işlem, ağ ve veri depolama bileşenlerinden oluşmaktadır. İşlem bileşenleri içerisinde yönetim, kenar ve misafir iş yüklerinin çalıştığı x86 tabanlı sunucular oturmaktadır. Bu tasarım önerilen mimariyi çalıştırmak için gereken fiziksel kabiliyetler için bir kılavuzluk yapsa da belirli bir donanım markası için tavsiyede bulunmamaktadır. Bütün bileşenler desteklenen yazılımlarla uyumlu donanımlar olmalıdır.

* + 1. Sanal Alt Yapı Katmanı

Sanal alt yapı katmanı fiziksel katman bileşenleri üzerinde yer almaktadır. Sanal alt yapı katmanı altta yatan fiziksel alt yapıya erişimi kontrol ederek yönetim ve misafir iş yüklerinin ihtiyaç duyduğu kaynakları tahsis eder. Yönetim iş yükü, bulut yönetim katmanı, servis yönetimi, iş sürekliliği ve güvenlik alanlarının elemanları ile birlikte sanal yönetim katmanının kendisi içerisinde bulunan elemanlardan oluşur.

* + 1. Bulut Yönetim Katmanı

Bulut yönetim katmanı yığının en üst katmanıdır. Servis tüketimi bu katmanda gerçekleşir. Bu katman kaynaklar için istekte bulunur ve daha alt katmanların eylemlerini organize eder. Bunu da en yaygın şekilde ya kullanıcı ara yüzü ile ya da uygulama programlama ara yüzü (API) ile sağlar. Her ne kadar SDDC yardımcı hizmetler olmadan da kendi başına ayakta durabilse de tam bir SDDC deneyimi için diğer destekleyici bileşenlere ihtiyaç duyulur. Hizmet yönetimi, iş sürekliliği ve güvenlik alanları, bu desteği sağlayarak mimariyi tamamlar.

* + 1. Servis ya da Hizmet Yönetimi

Herhangi bir tip IT alt yapısı inşa ederken, portföy ve operasyonlar yönetimi günlük hizmet temininde anahtar rol oynar. Bu mimarinin hizmet yönetimi alanı başlıca operasyon yönetimine (gözlemleme, uyarma ve log yönetim) odaklanmaktadır.

* + 1. Operasyon Yönetimi

Operasyon yönetimi katmanı mimarisi, bir SDDC içerisindeki esas tipteki operasyonlara destek sağlayan yönetim bileşenlerini içermektedir. Mikro segmentasyon kullanım örneğini vermek gerekirse gözlemleme ve loglama için vRealize Log Insight yazılımı ile sağlanabilmektedir.

Operasyonlar katmanında , alt tarafta bulunan fiziksel alt yapı,sanallaştırma yönetimi ve misafir iş yükleri gerçek zamanlı olarak gözlemlenmektedir. Gözlemlenen bu bilgiler yapılandırılmış veri (metrik) ve yapılandırılmamış veri (log) olarak iki biçimde toplanmaktadır. Operasyon katmanı aynı zamanda SDDC topolojisinden de haberdardır. Bu topoloji fiziksel ve sanal işlem, ağ ve veri depolamadan oluşmaktadır ki akılcı ve dinamik bir operasyon yönetimi için kritiktir. Operasyon katmanı, ana başlıklar olarak gözlemleme, loglama, yedek ve geri dönme, felaket kurtarma ve güvenliğe riayet ve uyumlulukdan oluşur. Bir arada bu bileşenler hizmet yönetimi, iş sürekliliği ve güvenlik gibi alanların birbirleri ile buluşmasını sağlarlar.

* + 1. İş Sürekliliği

Kurumsal bir sistem, veri yedekleme ve geri dönüşü ve felaket kurtarma sağlayarak iş sürekliliğini destekleyen bileşenler barındırmalıdır. Veri kaybı gerçekleştiğinde, kalıcı kayıpların gerçekleşmemesine engel olabilmek için doğru bileşenler konumlandırılmış olmalıdır. Bu tasarım yedekleme ve geri dönüşün nasıl konumlandırılıp işletileceği ve bir felaket durumunda yedek bileşenlere geçiş hakkında kılavuzluk sağlamaktadır.

* + 1. Güvenlik

Tüm sistemler tasarımdan itibaren güvenli olmalıdır. Güvenli bir tasarım riskleri azaltacağı gibi bir denetim yapısı sağlayarak uygunluğu artırır. Güvenlik alanı tüm SDDC’nin iç ve dış güvenlik tehditlerine karşı dirençli olması için nelerin gerekli olduğunu ana hatlarıyla çizmektedir.

* 1. Fiziksel Alt Yapı Mimarisi

Veri merkezi fiziksel katmanı mimarisi mantıksal donanım podlarına ve fiziksel ağ topolojisine dayanmaktadır.

* + 1. Pod Mimarisi

Yazılım tabanlı veri merkezi mimari tasarımı pod adı verilen küçük ortak yapı taşı setleri kullanmaktadır.

* + - 1. Pod Mimarisi Karakteristikleri

Podlar sunucular, veri depolama ekipmanları ve ağ ekipmanlarının değişik kombinasyonlarını içerebilir ve değişen seviyelerde donanım yedekliliği ve bileşen kalitesi sağlayacak şekilde kurulabilir. Podlar birbirleri arasında verileri dağıtabilen çekirdek bir ağa bağlıdırlar. Podlar bir SDDC ağ dokusu içerisindeki bağlı elemanların standart birimi olduklarından herhangi bir fiziksel özellik ile tanımlanmazlar.

Bir pod, SDDC platformu için işlevselliğin mantıksal sınırlarıdır. Genellikle her pod bir kabine yayılsa da birden fazla podu daha küçük kurulumlar ile tek bir kabin içerisinde bir araya getirebiliriz. Hem küçük hem de büyük kurulumlar düşünüldüğünde homojenlik ve kolay replikasyon önemlidir.

Aynı tipteki farklı podlar değişen gereksinimler yüzünden farklı karakteristikler sağlayabilmektedir. Örnek vermek gerekirse bir işlem podu yüksek erişilebilirlik için her bileşeni için tamamıyla donanım yedekliliği (güç kaynaklarından hafıza çiplerine kadar) sağlayabilir. Aynı zamanda aynı yapılandırmaya sahip bir başka işlem podu daha düşük maliyetli donanım yedekliliği gerektirmeyebilir. Bu değişkenlik sayesinde mimarimiz SDDC içerisindeki farklı iş yüklerine hizmet verebilir.

Bu tip kurulumlarda yol gösterici ilkelerden birisi de VLAN’lar tek bir podun ötesine ağ katmanı tarafından yayılmamalıdır. Her ne kadar bu VLAN sınırlaması basit bir gereksinim gibi görünse de fiziksel anahtarlama alt yapısının nasıl kurulacağı ve nasıl ölçekleneceği üzerinde çok büyük etkisi bulunmaktadır.

* + - 1. Pod ile Kabin Eşleştirmesi

Podlar veri merkezi kabinetleri ile bire bir örtüşmek zorunda değildirler. Bir pod tekrarlanabilir en küçük yapı taşı birimi iken, bir kabinet sadece bir boyut birimidir. Podlar birbirlerinden farklı boyutlarında olabildikleri için, podların nasıl veri merkezi kabinetlerine eşleştirilecekleri kullanımdan kullanıma farklılık gösterebilmektedirler.

|  |  |
| --- | --- |
| Bir kabinet içerisinde bir pod. | Bir pod tam olarak bir kabinet işgal eder. |
| Bir kabinet içerisinde birden fazla pod. | İki veya daha fazla pod bir kabinet işgal eder. Örnek vermek gerekirse bir yönetim podu, bir paylaşımlı kenar ve işlem podu aynı kabinet içerisinde yer alabilirler. |
| Birden fazla kabinet içerisine yayılmış tek bir pod. | Tek bir pod birden fazla komşu kabinetlere yayılabilir. Örnek vermek gerekirse bir veri depolama podu veya işlem podu birden fazla kabinete yayılmış olabilirler. |

* + 1. Pod Çeşitleri

SDDC, yönetim podu, işlem podu, kenar podu, paylaşımlı kenar ve işlem podu ve veri depolama podu olarak podları ayırmaktadır. Her tasarım çeşitli podlar içerir.

* + - 1. Yönetim Podu

Yönetim podu içerisinde SDDC’yi yöneten sanal makineler bulunmaktadır. Bu sanal makineler vCenter Server, vSphere Update Manager, NSX Manager, NSX Controller, vRealize Operations Manager, vRealize Automation, vRealize Log Insight ve diğer yönetim bileşenlerinden oluşmaktadır. Yönetim podu kritik alt yapıyı barındırdığı için bu pod için en azından basit de olsa donanım için bir yedeklilik planlanmalıdır.

Yönetim podları bileşenlerinde misafir özel adresleme kullanılmamalıdır.

* + - 1. Paylaşımlı Kenar ve İşlem Podu

Paylaşımlı kenar ve işlem podu, SDDC ve dış ağ arasındaki kuzey güney trafiğini yönlendirmek ve SDDC içerisindeki doğu batı trafiğini yönlendirmek için gereken NSX servislerini çalıştırır. Paylaşımlı pod aynı zamanda SDDC içerisindeki misafir sanal makineleri (iş yükü olarak da anılmaktadır) barındırır. SDDC büyüdükçe farklı tipteki iş yüklerini karşılamak ve farklı seviyelerdeki hizmet seviyesi anlaşmalarını destekleyebilmek için ek işlem podları eklenebilir.

* + - 1. İşlem Podu

İşlem podları, SDDC içerisinde misafir sanal makinelerini (iş yükü olarak da anılmaktadır) barındırır. Bir SDDC farklı tipteki işlem podlarını bir arada sunarak içerisinde farklı hizmet seviyeleri için ayrık işlem havuzları sağlayabilir.

* + - 1. Veri Depolama Podu

Veri depolama podları NFS veya iSCSI kullanarak ağ üzerinden erişilebilir veri depolama hizmeti sağlarlar. Farklı tipteki veri depolama podları, minimal veya hiç yedeklilik sağlamayan IDE disklerden oluşan JBOD’dan tutun da tümüyle yedekli kurumsal veri depolama birimleri ile farklı seviyelerdeki hizmet seviyesi anlaşmalarını destekleyebilir. IP tabanlı ve yüksek bant genişliğine ihtiyaç duyan veri depolama ihtiyaçları için bu podlar dinamik olarak ölçeklenebilir.

* + 1. Fiziksel Ağ Mimarisi

Bu sunduğumuz yazılım tabanlı veri merkezi tasarımı pek çok fiziksel ağ mimarilerinden istifade edebilir.

* + - 1. Ağ Transport

Bir SDDC için fiziksel anahtar katmanı oluştururken Layer 2 transport servisleri ya da Layer 3 transport servisleri kullanılabilir. Ölçeklenebilir ve üretici bağımsız bir veri merkezi için Layer 3 transport servisleri kullanılması tavsiye edilmektedir.

Tasarımımızda Layer 2 ve Layer 3 transportların her ikisi de desteklenmektedir. Layer 2 ya da Layer 3 kullanımına karar verirken aşağıdakiler göz önünde bulundurulmalıdır:

* NSX ECMP Edge cihazları yukarı yönlü Layer 3 cihazları ile, yönetim ve diğer iş yükü sanal makineleri için eşit maliyetli yönlendirme sağlayabilmek için yönlendirme komşuluğu oluştururlar.
* Mevcut fiziksel ağ alt yapısın için yapılmış yatırımlar.
* Layer 2 ve Layer 3 tasarımlarının avantajları ve dezavantajları.
  + - 1. Layer 2 Transport Avantajları ve Dezavantajları

Layer 2 transport kullanan bir tasarım için aşağıdaki durumların göz önüne alınması gerekmektedir:

* Layer 2 transport kullanan bir tasarımda, ToR anahtarlar ve yukarı yönlü Layer 3 cihazlar (çekirdek anahtarlar ya da yönlendiriciler) bir anahtar dokusu oluştururlar.
* Yukarı yönlü Layer 3 cihazlar her bir VLAN’ı sonlandırmalı ve varsayılan gateway özelliğini sağlamalıdır.
* ToR anahtarlardan yukarı yönlü Layer 3 cihazlara giden bağlantılar tüm VLAN’ları taşıyan 802.1Q trunkları olmalıdır.

Layer 2 transport kullanmanın aşağıdaki avantajları ve dezavantajları bulunmaktadır:

* Bu yaklaşımın faydası daha çok tasarım özgürlüğüdür. İstenirse VLAN’lar yayılabilir. Bu da bazı durumlarda kolaylık sağlar.
* En büyük çekince ise böyle bir uygulamanın büyüklüğü sınırlıdır. Tüm fabric elemanları sınırlı sayıda VLAN’ları paylaşmak zorundadır. Bunun yanı sıra tek bir üreticinin özelleşmiş veri merkezi anahtarlama ürününe mahkûm kalınabilir.
  + - 1. Layer 3 Transport Avantajları ve Dezavantajları

Layer 3 transport kullanan bir tasarım için aşağıdaki durumların göz önüne alınması gerekmektedir:

* Layer 2 bağlantı ToR anahtara kadar veri merkezi kabineti içerisinde sınırlıdır.
* Her bir VLAN’ı ToR anahtar sonlandırır ve gateway özelliğini sağlar. Dolayısıyla her bir VLAN için anahtar sanal ara yüzüne (SVI) sahiptir.
* ToR anahtarlarından yukarı yönlü katmanına olan bağlantılar noktadan noktaya yönlendirilen bağlantılardır. Bağlantılar üzerinde VLAN trunk tanımlanmasına izin verilmez.
* ToR anahtar, BGP, IS-IS veya OSPF gibi dinamik yönlendirme protokolleri ile yukarı yönlü anahtarlara bağlanırlar. Her kabinet içerisindeki her bir ToR genellikle her bir VLAN ya da subnet için bir adet küçük bir set önek yayınlarlar. Sırayla ToR, diğer ToR anahtarların yayınlayıp haberdar olduğu önekler için eşit maliyet yolları hesaplar.

Layer 3 transport kullanmanın aşağıdaki avantajları ve dezavantajları bulunmaktadır:

* Avantajı fiziksel anahtarlama çatısı için pek çok farklı Layer 3 uyumlu anahtar ürün seçilebilir. BGP, IS-IS ve OSPF uygulamalarının kendi aralarındaki uyumluluğu sayesinde farklı üreticilerin farklı ürünleri karma olarak kullanılabilir. Bu yaklaşım bahsi geçen fiziksel anahtarların sadece basit, temel yeteneklerini kullandığından genellikle daha maliyet etkindir.
* Layer 3 yönlendirme kullanımının çekincesi ve bir tasarım sınırlaması da VLAN’ların bir kabinet içerisinde sınırlı olmasıdır. Bu da vSphere Fault Tolerance ve veri depolama ağları kullanımını etkilemektedir. Bu sınırlandırma NSX’in Layer 2 köprüleme özelliği kullanılarak aşılabilir.
  + - 1. Alt Yapı Ağ Mimarisi

Ağ sanallaştırmanın ana hedeflerinden birisi sanal-fiziksel ağlar arasında soyutlamanın sağlanmasıdır. Bunu sağlayabilmek için fiziksel fabrik aşağıdaki özelliklere sahip olan sağlam bir IP transport yapısına sahip olmalıdır:

* Basitlik
* Ölçeklendirilebilirlik
* Yüksek bant genişliği
* Hata dirençli transport
* Çeşitli seviyelerdeki hizmet kalitesi desteği (QoS)
  + - 1. Basitlik ve Ölçeklendirilebilirlik

Ağ üzerinde basitlik ve ölçeklendirilebilirlik ilk ve en önemli gereksinimdir.

* + - 1. Basitlik

Bir veri merkezi içerisindeki anahtarların yapılandırmaları basit olmalıdır. AAA, SNMP, syslog, NTP ve buna benzer servislerin genel ya da global yapılandırmaları ağ anahtarlarının yerleşiminden bağımsız olarak satır satır çoğaltılabilmelidir. Tüm ağ anahtarların tek bir seferde merkezi bir yönetim vasıtası ile yapılandırılması da bir başka alternatiftir.

Çoklu şasi bağlantı birleşim grupları, VLAN ID’leri ve dinamik yönlendirme protokol yapılandırmaları gibi ağ anahtarlarına özgün yapılandırmalar minimumda tutulmalıdırlar.

* + - 1. Ölçeklendirilebilirlik

Ölçeklenebilirlik faktörleri aşağıdakileri içerir:

* Bir fabrik içerisindeki desteklenen kabinet sayısı.
* Bir veri merkezi içerisindeki iki kabinet arasındaki bant genişliği miktarı.
* Kabinetler arası yol adedi.

Tüm ağ anahtarları üzerinde müsait olan bağlantı portların toplam sayısı ve kabul edilebilir fazla taahhüt miktarı bir fabrik içerisinde desteklenen kabinet sayısını belirler. Farklı kabinetler farklı tipte alt yapı barındırabileceği için farklı bant genişliği gereksinimleri olur.

* İçerisinde IP tabanlı veri depolama sistemleri içeren kabinetler diğer kabinetlerden daha fazla ağ trafiğine sahiptirler.
* Üzerinde sanal makine ya da iş yükleri bulunan hipervizörler barındıran işlem kabinetlerinin paylaşımlı kenar ve işlem kabinetlerinden farklı bant genişliği gereksinimleri bulunur.
  + - 1. Fiziksel Ağ Ara Yüzleri (NIC)

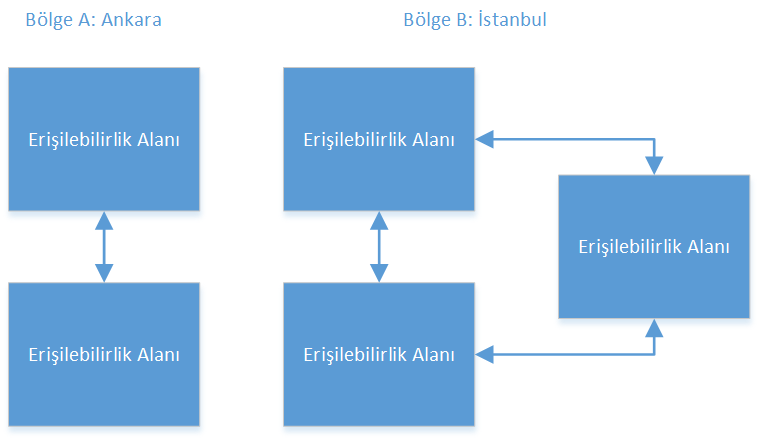
Eğer bir sunucunun aynı hıza sahip birden fazla ağ ara yüz kartı varsa, bu iki kart VLAN’ların trunklandığı ortak bağlantılar olarak kullanılmalıdır.

vSphere Distributed Switch pek çok değişik NIC birleştirmesini desteklemektedir. Yüke bağlı NIC birleştirmesi hem bant genişliğinin optimal kullanımını sağlar hem de herhangi bir bağlantı arızasına arşı yedeklilik sağlar. Her sunucu üzerinde en az iki adet 10GbE kart kullanılarak yaprak ağ anahtarlarına bağlanılmalıdır. Bu ağ anahtarları 802.1Q az sayıdaki VLAN’ların trunklanmasını desteklemelidir. Bu trafiklere örnek vermek gerekirse: veri depolama, VXLAN, vSphere Replication ve vMotion vs.

* + 1. Erişilebilirlik Alanları ve Bölgeler

Bir SDDC içerisinde, erişilebilirlik alanları, alt yapı bileşenlerinin bir koleksiyonudur. Bölgeler ise felaket kurtarma çözümlerini destekler ve iş yüklerini son kullanıcılara yakın yerlere konumlandırırlar. Genellikle birden çok erişilebilirlik alanı bir arada bir bölgeyi oluştururlar.

Tasarımımızda iki bölge ve her bir bölge içerisinde tek bir erişilebilirlik alanı kullanılmıştır. Aşağıdaki çizimde tasarımın nasıl birden çok erişilebilirlik alanı içerecek şekilde genişletilebileceği gösterilmiştir.



* + - 1. Erişilebilirlik Alanları

Her erişilebilirlik alanı, arızların veya kesintilerin alan sınırlarını aşarak yayılmasını engellemek amacıyla diğerlerinden yalıtılmıştır.

Bir arada birden çok erişilebilirlik alanları yedeklilik sayesinde sürekli kullanılabilirlik sağlayarak kesintilere engel olur ve hizmet seviyesi anlaşmalarının geliştirilmesini sağlarlar. Harici etmenlerin sebep olduğu kesintiler sadece bir alanı etkilemelidir. Büyük felaketleri göz ardı edersek bu etmenler diğer alanlarda kesintiye yol açmazlar. Her bir erişilebilirlik alanı güvenilir olması için tasarlanmış bağımsız alt yapısına sahip kendi fiziksel bölgesi içerisinde çalışırlar. Her bir alanın bağımsız güç kaynaklarına, soğutma sistemine, ağ alt yapısına ve güvenliğe ihtiyacı vardır. Bir veri merkezi içerisindeki kesintiye yol açan yaygın arıza sebeplerinden olan jeneratörler ve soğutma ekipmanları erişilebilirlik alanları arasında paylaşılmamalıdır. Ek olarak bu alanlar sıra dışı felaketlere maruz olmamaları için fiziksel olarak ayrı olmalıdırlar. Erişilebilirlik alanlar genellikle ya birbirlerine makul ölçüde yakınlıkta (ağ gecikme süreleri tek basamağı geçmemeli) olan iki ayrık veri merkezi şeklinde ya da büyük ölçekli bir veri merkezi içerisinde iki ayrı bölüm içerisinde olmalıdırlar.

Tek bir bölgeye ait birden çok erişilebilirlik alanı olabilir. Erişilebilirlik alanları arasındaki mesafe yaklaşık olarak 50 kilometreye kadar olabilirler. Dark fiber kullanımı sayesinde iki alan arasındaki ağ gecikmesi tek haneli rakamlarda tutulabilir. Bu mimari SDDC ekipmanının aktif/aktif olarak alanlar arasında tek bir veri merkeziymişçesine kullanımına müsaade eder.

İş yükleri aynı bölgedeki birden çok erişilebilirlik bölgesi arasında tek bir sanal veri merkezinde çalışıyormuş gibi işletilebilir. Bu da kritik uygulamaların çalışmasına uygun oldukça yüksek bir erişilebilirlik sağlayan mimariyi destekler. İki yer arasındaki mesafe çok uzak olmaya başladığında bu yerler artık iki erişilebilirlik alanı olarak görevlerini yerine getiremezler, bu yerlerin iki yarı bölge olarak değerlendirilmesi gerekir.

* + - 1. Bölgeler

Birden fazla bölge kullanarak iş yüklerini, son kullanıcıların fiziksel olarak yakın oldukları yerlere konumlandırabiliriz. Örnek vermek gerekirse, ABD batı yakasında bir bölge, doğu yakasında bir başka bölge olabilir ya da ABD bir bölge, Avrupa bir başka bölge olabilir.

Bölgeler çeşitli şekillerde yararlıdır:

* Bölgeler felaket kurtarma çözümlerini desteklerler. Bir bölge ana site olabilirken diğer bir başkası kurtarma sitesi olabilir.
* Bazı ülkelerin veri gizliliği hakkındaki yasaları ve sınırlandırmalarına uyabilmek amacıyla birden fazla bölge oluşturup, her kullanıcının bilgilerini ve iş yüklerini kendi ülkesinin sınırları içerisindeki bölgede tutulabilir.

Bölgeler arasındaki mesafe oldukça büyük olabilir. Bu tasarımda iki adet örnek bölge kullanılmıştır. Bu bölgelerden birisi Ankara diğeri de İstanbul’dur.

* 1. Sanal Alt Yapı Mimarisi

Sanal alt yapı işlevsel bir SDDC’nin temelidir. Sanal alt yapı katmanları içerisinde fiziksel alt yapı kontrol edilir ve yönetim ve diğer iş yükleri için tahsis edilir. Sanal alt yapı katmanı başlıca fiziksel sanallaştırma sunucularının hipervizörlerden ve bu hipervizörlerin kontrolünden oluşur. Yönetim iş yükleri sanal yönetim katmanı içerisindeki elemanlardan oluşur. Bu elemanların yanı sıra bu katmanda bulut yönetimi katmanı ve hizmet yönetimi, iş sürekliliği ve güvenliğe dair elemanlarda bulunmaktadır.

* + 1. Yönetim Podu

Yönetim podu içerisinde SDDC’yi yöneten sanal makineler bulunmaktadır. Bu sanal makineler vCenter Server, vSphere Update Manager, NSX Manager, NSX Controller, vRealize Operations Manager, vRealize Automation, vRealize Log Insight ve diğer yönetim bileşenlerinden oluşmaktadır. Tüm yönetim, gözlemleme ve alt yapı servisleri bu kritik hizmetlere yüksek erişilebilirlik sağlayan bir vSphere sunucu kümesine konumlandırılır. Yönetim sunucu kümesine üzerinde tanımlı izinler sadece yöneticilerin buraya erişmesi şeklinde sınırlandırılmıştır. Bu sınırlandırma yönetim, gözlemleme ve alt yapı servislerini çalıştıran sanal makineleri korumak içindir.

* + 1. Ağ Sanallaştırma Bileşenleri

SDDC mimarisi içinde, ağ sanallaştırma platformu olan VMware NSX önemli bir bileşendir. NSX platformu ağ sanallaştırma tasarımı ile ilgili olan birçok bileşenden oluşmaktadır.

* + - 1. NSX Platformu

NSX for Vsphere bir ağ sanallaştırma katmanı oluşturur. Tüm sanal ağlar, fiziksel ve sanal ağlar arasında bir soyutlama olan bu katmanın üzerine kurulur. Bu ağ sanallaştırma katmanını oluşturmak için çeşitli bileşenler gerekmektedir.

* vCenter Server
* NSX Manager
* NSX Controller
* NSX Virtual Switch

Bu bileşenler, iletişim sınırları oluşturmak ve iş yükleri ile sistem kontrol mesajlarının arasında izolasyon sağlayabilmeleri için değişik düzlemlere ayrılmışlardır.

|  |  |
| --- | --- |
| Veri Düzlemi | İş yükü verileri tamamı ile veri düzlemi içerisinde kapsanmıştır. NSX mantıksal anahtarları birbirleri ile ilgisi olmayan iş yükü verilerini ayırır. Veri, fiziksel ağ üzerinde bu işe tayin edilmiş iletişim ağları vasıtası ile taşınır. NSX Virtual Switch, dağıtık yönlendirme ve dağıtık güvenlik duvarı da bu düzlemde uygulamaya alınır. |
| Kontrol Düzlemi | Ağ sanallaştırmasının kontrol mesajları kontrol düzlemi içerisinde yer alır. Kontrol düzlemi iletişimleri, veri düzlemi için kullanılan taşıma ağlarından izole edilmiş güvenli fiziksel ağlar (VLAN) vasıtasıyla yapılmalıdır. Kontrol mesajları, NSX Virtual Switch oluşumlarının özelliklerinin yapılandırılması ve her bir ESXi sunucudaki dağıtık güvenlik duvarı yapılandırmaları için kullanılır. |
| Yönetim Düzlemi | Ağ sanallaştırması orkestrasyonu yönetim düzleminde gerçekleşir. Bu katmanda vRealize Automation gibi bulut yönetim platformları sanal iş yükleri için ağ kaynaklarına istekte bulunabilir, tüketebilir ve kaldırabilir. İletişim bulut yönetim platformu tarafından sanal makineler oluşturulup yönetilebilmesi için ve NSX Manger’ın ağ kaynaklarını kullanabilmesi için vCenter’a yönlendirilir. |

* + 1. Ağ Sanallaştırma Servisleri

Ağ sanallaştırma servisleri, mantıksal anahtarları, mantıksal yönlendiricileri, mantıksal güvenlik duvarlarını ve NSX’in diğer bileşenlerini içerir.

* 1. Bulut Yönetim Platform Mimarisi
     1. Bulut Yönetimi Platformu için vRealize Automation Mimarisi
     2. vRealize Business for Cloud Mimarisi
  2. Operasyonlar Mimarisi
     1. Operasyonlar Yönetim Mimarisi
     2. Loglama Mimarisi
     3. Veri Koruma ve Yedekleme Mimarisi
     4. Felaket Kurtarma Mimarisi
     5. vSphere Update Manager Mimarisi

1. Detaylı Tasarım

Yazılım tanımlı veri merkezi (SDDC) detaylı tasarımı hem fiziksel hem de sanal alt yapı tasarımlarını içerir. İçerisinde tasarım kararları ve bu kararların gerekçeleri bulunmaktadır.

Ayrıca her bölüm konu hakkında tartışma ve çizimleri içermektedir.

|  |  |
| --- | --- |
| Fiziksel Alt Yapı Tasarımı | Her veri merkezinin temelini oluşturan üç ana başlık üzerinde odaklanır. Bu bölümde erişilebilirlik bölgeleri ve alanlar hakkında bilgiler bulunmaktadır. Bu bölümde ayrıca kabin ve pod yapılandırmaları, fiziksel sunucular, bağlı oldukları veri depolama birimleri ve ağ yapılandırmaları hakkında ayrıntılı bilgiler bulunmaktadır. |
| Sanal Alt Yapı Tasarımı | Özdeki sanallaştırma yapılandırmasına dair detaylı içerir. Bu bölümde ESXi hipervizörü, vCenter Server, NSX’in de dahil olduğu sanal ağ tasarımı ve veri depolama sanallaştırması vSAN hakkında bilgiler bulunur. İş sürekliliği (yedekleme ve geri dönüş) ve felaket kurtarma hakkında detaylar burada verilir. |
| Bulut Yönetim Platformu Tasarımı | vRealize Automation ve vRealize Orchestrator’u kullanan SDDC yığının tüketim ve orkestrasyon katmanı hakkında bilgiler içerir. IT organizasyonları tamamıyla dağıtık ve ölçeklendirilebilir mimari ile provizyonlama ve hizmetten kaldırma işlemlerini gerçekleştirebilirler. |
| Operasyonlar Alt Yapı Tasarımı | Operations Manager ve Log Insight yazılımlarının mimarilerinin nasıl oluşturulacağından, nasıl kurulacakları ve yapılandırılacaklarından bahseder. |

* 1. Fiziksel Alt Yapı Tasarımı

Fiziksel alt yapı tasarımı erişilebilirlik alanları ve bölgeleri, veri merkezi kabinetleri üzerinde pod yerleşimine dair tasarım ayrıntılarını içermektedir.

Sunucu, ağ ve veri depolama donanımlarına dair tasarım kararları fiziksel alt yapı tasarımının bir parçasıdır.

* + 1. Fiziksel Tasarım Temelleri

Fiziksel tasarım temelleri erişilebilirlik alanları ve bölgeler, pod çeşitleri ve kabinetler hakkındaki kararları içerir. ESXi host fiziksel tasarımı da tasarım temelleri içinde yer alır.

* + - 1. Erişilebilirlik Alanları ve Bölgeler

Erişilebilirlik alanları ve bölgeler değişik amaçlar için kullanılırlar

|  |  |
| --- | --- |
| Erişilebilirlik Alanları | Bir erişilebilirlik alanı, SDDC’nin birim hata bölgesidir. Bir SDDC içerisindeki birden fazla erişilebilirlik bölgesi olması, SDDC’ye sürekli erişilebilirlik sağlar ve hizmet kesintisi ihtimalini en aza indirir. |
| Bölgeler | Bölgeler, farklı SDDC oluşumları arasında felaket kurtarma olanağı sağlarlar. Bu tasarımda iki bölge kullanılmıştır. Her bölge ayrı bir SDDC oluşumudur. Bölgelerin benzer fiziksel katman tasarımı ve sanal alt yapı tasarımı fakat farklı isimlendirmeleri bulunur. |

Bu tasarımda iki bölge ve her bir bölge içerisinde tek bir erişilebilirlik alanı bulunmaktadır.

|  |  |  |  |
| --- | --- | --- | --- |
| Bölge | Bölge Kodu | Bölgeye Özel Ad | Bölge Tanımlaması |
| A | ANK01 | ank01.kurum.gov.tr | Ankara, Merkez |
| B | KON01 | kon01.kurum.gov.tr | Konya, FKM |

Tablo 1 Erişilebilirlik Alanları ve Bölge Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-FİZ-001 | Bölge başına, tüm SDDC yönetimini destekleyecek tek bir erişilebilirlik alanı oluşturmak. | Tek bir erişilebilirlik alanı, bir bölge için gerekli tüm SDDC yönetim ve işlem bileşenlerini destekleyebilir. Daha sonra yeni bir erişilebilirlik alanı ile SDDC’yi genişletip, ölçeklendirilebilir. | Tüm çözüm için sınırlı bir yedeklilik sağlar. Tek bir erişilebilirlik alanı tek hata noktası haline gelebilir. |
| SDDC-FİZ-002 | İki bölge oluşturmak. | Tasarım hedeflerinde tarif edilen çoklu bölge yük devretme teknik gereksinimini karşılamaktadır. | Birden çok bölge oluşturmak çözümün büyümesine ve maliyetinin artmasına yol açar. |

* + - 1. Podlar ve Kabinetler

SDDC fonksiyonları birden fazla podlar arasında bölüştürülmüştür. Her pod bir ya da birden fazla kabineti işgal edebilir. Podların toplam adetleri ve çeşitleri ölçeklenebilme ihtiyacına göre değişiklik gösterebilir.

Tablo 2 - Gerekli Kabin Sayısı

|  |  |  |  |
| --- | --- | --- | --- |
| Pod (Fonksiyon) | Gerekli Kabinet Sayısı (tam ölçekli yapılandırma) | Minimum Kabinet Sayısı | Yorum |
| Yönetim ve Paylaşımlı kenar ve işlem podu | 1 | 1 | İki yarım kabinet yönetim podu ve paylaşımlı kenar ve işlem podu için yeterlidir. Sayıları ve kaynak ihtiyaçları arttıkça küme sunuculara yenilerinin eklenmesi gerekecektir ve kabinet içerisindeki boş alanlar bunlar için ayrılmalıdır. |
| İşlem podu | 3 | 0 | Her birinde 24 ESXİ sunucular olan 3 Kabinet ve 3 işlem podu ile hedef rakam olan ortalama boyutlardaki 3000 sanal sunucuya erişilir. Sanal sunucularda iki vCPU ve 4GB RAM bulunduğu var sayılırsa ve ani yükler için 20% pay ile 70 sunucu gereklidir.  Miktarlar ve performans gereksinimleri işlem podu içerisindeki iş yüklerine göre değişiklik gösterir |
| Veri depolama podu | 3 | 0 (Eğer VSAN kullanılacaksa) | VSAN olmayan veri depolama podları izole ayrı kabinetlerde bulunurlar. |
| Toplam | 7 | 1 |  |

Tablo 3 - Pod ve Kabinet Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-FİZ-003 | Yönetim ve Paylaşımlı kenar ve işlem podu aynı kabinet içerisinde yer alırlar. | Yönetim podu (4 ESXi sunucu) ve paylaşımlı kenar ve işlem podu (4 ESXi sunucu) sayısı her biri için ayrı kabinet gerektirmeyecek kadar azdır. Fiziksel ağlarla olan bağlantı bu kabinet üzerinden sağlanabilir. Yönetim ve işlem için olan kenar bağlantı kaynaklarını bir arada bulundurarak VLAN yayılması en aza indirilir. | Tasarım yeterli enerji ve soğutma kaynaklarını içermelidir. Eğer bu kabinet tümden devre dışı kalırsa kesintiye maruz kalmamak için bir başka bölgeye yükün devredilmesi gerekebilir. |
| SDDC-FİZ-004 | Veri depolama podları birden fazla kabinete yayılabilirler. | SDDC’nin yatay olarak ölçeklenebilmesi adına veri depolama podu ve kabinet ilişkisi standart olmalıdır. | Tasarım yeterli enerji ve soğutma kaynaklarını içermelidir. |
| SDDC-FİZ-005 | Her kabinet iki ayrı enerji kaynağından beslenmelidir. | Güç yedekliliği, enerji kesintilerinde bir kabinet içerisindeki tüm ekipmanın birden kesilmesini engeller.  Yedekli ağ bağlantıları ile birlikte güç yedekliliği tüm bir kabinette gerçekleşebilecek arızalara engel olur. | Bütün cihazlar en az iki güç girişine sahip olmalı ve bu enerji girişlerinden birisinde enerji kesilirse çalışmaya devam edebilmelidir. |
| SDDC-FİZ-006 | Yönetim podu için gereken işlem kaynakları (en az 4 ESXi sunucu) kabinet içerisinde bir arada yerleştirilmelidir. | Yönetim podu için gereken işlem kaynaklarının bir arada yerleştirilmesi fiziksel veri merkezi tasarımını, konumlandırılmasını ve hata gidermeyi kolaylaştırır. | Yok. |
| SDDC-FİZ-007 | Paylaşımlı kenar ve işlem podu için gereken kaynakları (en az 4 ESXi sunucu) kabinet içerisinde bir arada yerleştirilmelidir. | Paylaşımlı kenar ve işlem podu için gereken işlem kaynaklarının bir arada yerleştirilmesi fiziksel veri merkezi tasarımını, konumlandırılmasını ve hata gidermeyi kolaylaştırır. | Yok. |

* + - 1. ESXi Sunucu Fiziksel Tasarım Teknik Özellikleri

ESXi sunucu fiziksel tasarım teknik özellikleri bu tasarımın kurulumunda ve test edilmesi esnasında kullanılan sunucuların karakteristik özelliklerini içerir.

Her sistemin yapılandırılması ve kurulumu işlemleri standartlaştırılmalıdır. Tüm bileşenler her bir sunucu üzerinde aynı şekilde kurulmalıdır. Standartlaşma değişkenliğin önüne geçtiği için yönetilebilir ve desteklenebilir bir alt yapı için kritik öneme sahiptir. Özellikle ağ kartları gibi PCI kartlarının tutarlı olarak hep aynı girişlere takılmış olması fizikselden sanala giden kaynakların uyumlu olarak yapılandırılması elzemdir. Tüm ESXi sunucuları birbirleri ile ağ, veri depolama için aynı şekilde yapılandırılmalıdır. ESXi sunucular VMware uyumluluk listesinde bulunan sunucular olmalı ya da listede bulunan onaylı donanım bileşenlerinden oluşmalıdır.

Yönetim ve kenar podları için kullanılacak ESXi sunucuların fiziksel sunucuların seçilmesi özel gereksinimlere tabiidir. Bu sunucular ayrıca VMware Virtual SAN Ready Nodes listesinde bulunmalı ya da bu listedeki donanım bileşenlerinden seçilmiş olmalıdır.

* Ortalama bir sanal sunucu 2 vCPU ve 4GB RAM’a sahiptir
* Standart 2U yüksekliğindeki bir sunucu ESXi sunucu olarak ortalama 60 adet sanal sunucuyu barındırabilir.

Tablo 4 - ESXi Sunucu Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-FİZ-08 | VSAN Ready Node kullanılmalıdır | VSAN ready node kullanarak VSAN uyumluluğu garanti edilmiş olunur. | Donanım seçeneklerini sınırlandırabilir. |
| SDDC-FİZ-09 | Bir kümeye dahil olan tüm sunucularda aynı yapılandırma bulunmalıdır. | Dengeli bir küme donanım arızaları sırasında öngörülebilir performans sağlar. Bunun yanı sıra resync/rebuild işlemleri sırasında küme dengeli ise performansa etkisi daha az olur. | Satın alma süreçleri ve üretici seçimi küme başına düşünülmelidir. |

* + - 1. ESXi Sunucu Hafızası

İşlem podları için gerekli olan hafıza miktarı o pod üzerinde çalışan iş yüküne göre değişiklik gösterecektir. İşlem podları için hafıza miktarını belirlerken önemli olan nokta kabul denetim servisinin (admission control) arıza ya da bakım için bir sunucuyu ayırdığı n+1 ayarıdır.

Tablo 5 - ESXi Hafıza Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-FİZ-010 | Yönetim podu içerisindeki tüm ESXi sunucularda en az 192GB hafıza bulunmalıdır. | Bu pod içerisindeki yönetim ve kenar sanal sunucular için toplamda 424GB hafıza gerekmektedir. | Yok. |

Bir ESXi sunucu üzerinde SCSI tabanlı cihazlar (SAS/SATA/SAN) kullanırken minimum boot disk boyutu en az 5GB’dır. ESXi sunucular yerel SAN SCSI boot cihazları kullanılarak ya da vSphere Auto Deploy kullanılarak açılabilirler.

Neyin desteklendiği VSAN sürümüne göre farklılık gösterebilir:

* VSAN durum bilgisiz (stateless) vSphere Auto Deploy’u desteklememektedir.
* VSAN 5.5 ve üstü sürümler USB/SD kart gibi gömülü cihazları 4GB dan büyük olmak kaydıyla ESXi sunucular için boot cihazı olarak desteklemektedir.
* VSAN 6.0’dan itibaren SATADOM’da boot cihazı olarak desteklenmektedir.
  + 1. Fiziksel Ağ Tasarımı

Burada anlatılan yazılım tanımlı veri merkezi tasarımı kurumsal seviye fiziksel ağ mimarilerinin çoğunu kullanabilir.

* + - 1. Anahtar Tipleri ve Ağ Bağlantıları

Fiziksel ortamın kurulu dikkatli bir değerlendirme gerektirmektedir. Fiziksel anahtarlar, anahtar bağlantıları, VLAN’lar ve alt ağları ve erişim noktaları için tavsiye edilen en iyi uygulama yöntemleri kullanılmalıdır.

* + - 1. Kabin Üstü (Top of Rack TOR) Fiziksel Anahtarlar

TOR anahtarların yapılandırılmasında aşağıdaki en iyi uygulama yöntemler dikkate alınmalıdır:

* Erişilebilirliğin artırılabilmesi için yedekli fiziksel anahtarlar kullanılmalıdır.
* ESXi sunucu bağlantı noktalarına bağlanan anahtar bağlantı noktaları el ile trunk bağlantı noktası olarak yapılandırılmalıdırlar. Sanal anahtarlar pasif cihazlar olup DTP gibi dinamik trunklama protokollerini alıp geçirmezler.
* ESXi ağ kartlarının bağlı olduğu tüm bağlantı noktalarında STP protokolünü düzenlenmelidir. Bu sayede geçiş yapan bağlantı noktalarında bu geçişin süresi düşürülebilmektedir. Örnek olarak Cisco ağ anahtarlında Trunk PortFast ayarının etkinleştirilmesi gibi.
* Yönetim ve VXLAN VMkernel bağlantı noktaları tarafından kullanılan tüm VLAN’larda DHCP ve DHCP Helper kabiliyetleri sağlanmalıdır. Bu sayede yapılandırmak esnasında ilgili alt ağa ait IP yapılandırması DHCP ile kolayca yapılabilir.
* Tüm anahtar bağlantı noktalarında, anahtarların kendi aralarındaki bağlantılarda ve anahtarlanan sanal ara yüzlerde (switched virtual interfaces, SVI) jumbo frame özelliği açılmalıdır.
  + - 1. Kabin Üstü Anahtar Bağlantıları ve Ağ Ayarları

Tüm ESXi sunucuları SDDC ağ dokusuna iki adet 10GbE bağlantı noktası vasıtasıyla ToR anahtarlar ile bağlanırlar. ToR anahtarlar 802.1Q trunk ile tüm gerekli VLAN’ları temin edecek şekilde yapılandırılmalıdır. Bu yedekli bağlantılar bir ether-channel’ın (LAG/vPC) parçaları olmamakla birlikte vSphere Distributed Switch ve NSX’in özelliklerini kullanarak hiçbir ara yükün aşırı yüklenmemesini ve ayakta oldukları sürece tüm yedekli yolların kullanımını garantilerler.

* + - 1. VLAN’lar ve Alt Ağlar

Tüm ESXi sunucular VLAN’ları ve karşılık gelen alt ağları kullanırlar. Aşağıdaki esaslara dikkat edilmelidir:

* Karışıklığı ve IPv4 alt ağ bölümlendirmesi ile ilgili hataları azaltmak adına mümkün olduğunca /24 bit alt ağlar kullanılmalıdır.
* Virtual Router Redundancy (VRRP) ya da Hot Standby Routing Protocol (HSRP) için .253 IP’si (değişen olarak), .251 ve .252 IP’leri kullanılmalıdır.
* Bu alt ağlar için RFC1918 IPv4 adres uzayı kullanılmalıdır ve bir oktet böğle için bir başka oktet ise fonksiyon için ayrılmalıdır. Örnek olarak 172.bölgekodu.fonksiyon.0/24 aşağıdaki gibi bir sete tekabül etmektedir: (verilen VLAN’lar ve IP aralıkları örnek olarak verilmiştir uygulamada değişiklik gösterebilirler)

Tablo 6 - VLAN'lar ve IP Aralıkları Örneği

|  |  |  |  |
| --- | --- | --- | --- |
| Pod | Fonksiyon | Örnek VLAN | Örnek IP aralığı |
| Yönetim | Yönetim | 1611 (Özgün) | 172.16.11.0/24 |
| Yönetim | vMotion | 1612 | 172.16.12.0/24 |
| Yönetim | VXLAN | 1613 | 172.16.13.0/24 |
| Yönetim | vSAN | 1614 | 172.16.14.0/24 |
| Paylaşımlı Kenar ve İşlem | Yönetim | 1631 (Özgün) | 172.16.31.0/24 |
| Paylaşımlı Kenar ve İşlem | vMotion | 1632 | 172.16.32.0/24 |
| Paylaşımlı Kenar ve İşlem | VXLAN | 1633 | 172.16.33.0/24 |
| Paylaşımlı Kenar ve İşlem | vSAN | 1634 | 17216.34.0/24 |

* + - 1. Erişim Bağlantı Noktası Ağ Ayarları

Yaprak (leaf) anahtarları karşılık gelen sunuculara bağlayan noktalarda aşağıdaki ek ağ yapılandırmaları gerçekleştirilmelidir:

|  |  |
| --- | --- |
| Spanning-Tree Protokolü | Her ne kadar bu tasarım STP kullanmasa da ağ anahtarları genelde varsayılan olarak STP açık gelmektedirler. Erişim bağlantı noktalarını trunk PortFast olarak yapılandırılmalıdır. |
| Trunklama | VLAN’lar 801.1Q trunk üyeleri olarak yapılandırılmalı ve yönetim VLAN’ı da özgün VLAN olmalıdır. |
| MTU | Tüm VLAN’ları ve SVI’ların MTU’ları (yönetim, vMotion, VXLAN ve veri depolama) tutarlılık amacıyla jumbo frame olacak şekilde yapılandırılmalıdır. |
| DHCP Helper | Yönetiminin VIF’i, vMotion ve VXLAN alt ağı bir DHCP vekili olacak şekilde yapılandırılmalıdır. |
| Multicast | Tüm kabin üstü anahtarlarda IGMP gözetlemesi açılmalı ve her bir VLAN’da IGMP sorgulayıcı tanımlanmalıdır. |

* + - 1. Bölgeler Arası Bağlantı

İki bölgenin SDDC yönetim ağları, VXLAN kernel bağlantı noktaları ve kenar ve işlem VXLAN kernel bağlantı noktaları birbirlerine bağlı olmalıdır. Bu bağlantılar VPN üzerinden, noktadan noktaya devreler üzerinde de olabilirler. Son kullanıcılar her iki bölgenin dışa bakan ağ bölümlerine erişebilmelidirler.

Bölgeler arası bağlantı jumbo frameleri desteklemeli ve gecikme en fazla 150ms olmalıdır.

Bölgeler arası bağlantı çözümü bu tasarımın ve tezin konusu ve kapsamı dışındadır.

* + - 1. Fiziksel Ağ Tasarım Kararları

Fiziksel ağ tasarım kararları, VLAN’ların fiziksel yerleşimi ve kullanımını idare eden kararlardır. Jumbo frame kullanımı kararları ve DNS ve NTP gibi bazı ağ ilişkili gereksinimlere dair kararları içerir.

Tablo 7 - Fiziksel Ağ Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-FİZ-AĞ-001 | Fiziksel ağ mimarisi aşağıdaki gereksinimleri karşılamalıdır:   * ESXi sunucular üst bağlantıları için her bir ToR anahtar üzerinde 1 adet 10GbE bağlantı bulunmalıdır. * Sunucu üst bağlantıları etherchannel (vPC ya da LAG) olarak yapılandırılmamalıdır. * BGP’yi destekleyen Layer 3 cihazlar * IGMP desteği | ESXi sunucu başına iki üst bağlantı, anahtar arızaları esnasında erişilebilirliği garanti eder.  Bu tasarım bağlantı birleştirme (link aggregation) teknikleri ile uyumlu olmayan vSphere Distributed Switch, NSX ve çekirdek vSphere platformunu kullanır.  Bu tasarımda dinamik yönlendirme protokolü olarak BGP seçilmiştir.  NSX Hybrid kip replikasyonu IGMP’ye ihtiyaç duyar. | Donanım seçeneklerini daha sınırlıdır.  Fiziksel ağ yığınında dinamik yönlendirme protokollerinin yapılandırılması gerekmektedir. |
| SDDC-FİZ-AĞ-002 | BGP yönlendirme komşuluğu yapılandırılmış bir fiziksel ağ kullanılmalıdır. | Tasarım dinamik yönlendirme protokolü olarak BGP’yi tercih etmektedir. Bu sayede çoklu kiracı ve çoklu site kullanımına imkân verir. | Fiziksel ağ yığınında BGP yapılandırması gerekir. |
| SDDC-FİZ-AĞ-003 | Her kabinde iki adet ToR anahtar bulunmalıdır. Bu anahtarlar her bir sunucuya iki 10GbE bağlantı sağlamalıdır. | Bu tasarım yedeklilik sağlamak ve tasarım kompleksliğini azaltmak için iki adet 10 GbE kullanır. | Her kabin içerisinde iki adet ToR anahtar gerektirir ve bu da maliyetleri artırabilir. |
| SDDC-FİZ-AĞ-004 | Fiziksel ağ fonksiyonlarını bölümlendirebilmek için VLAN’lar kullanılmalıdır. | Çok miktarda NIC gerektirmeden fiziksel ağ bağlantıları sağlanmış olur.  SDDC içerisindeki farklı ağ fonksiyonları için ayrıştırmak gereklidir. Bu ayrışma farklılaşmış hizmetlere ve trafiğin önceliklendirilebilmesine imkân verir. |  |

* + - 1. Ek Tasarım Kararları

Ek tasarım kararları statik IP adresleri, DNS kayıtları ve gerekli NTP zaman kaynakları ile ilgili kararlardır.

Tablo 8 - Ek Ağ Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-FİZ-AĞ-005 | SDDC alt yapısındaki tüm yönetim bileşenlerine statik IP’ler atanmalıdır. Tek istisna DHCP ile yapılandırılan NSX VTEP’leridir. | Statik IP yapılandırması ile DHCP servisinin kesintilerinden kaynaklanabilecek sorunlardan ya da yapılandırma hatalarından korunmuş olur. | IP adres yönetimi oluşturulmuş olmalıdır. |
| SDDC-FİZ-AĞ-006 | Tüm yönetim birimleri için düz, ters, kısa ve FQDN çözümleyici DNS kayıtları oluşturulmalıdır. | Yönetim birimlerinin IP adresleri ve isim çözümlemesinin tutarlı bir şekilde gerçekleşmesini sağlar. | Yok. |
| SDDC-FİZ-AĞ-007 | Tüm yönetim birimleri için bir ortak NTP zaman kaynağı kullanılmalıdır. | Yönetim birimleri arasında zaman senkronizasyonu olması kritiktir. | Yok. |

* + - 1. Jumbo Frames Tasarım Kararları

IP veri depolama verimliliği Jumbo frame yapılandırmasından faydalanabilir. Frame başına yükün 1500 byte’dan Jumbo frame’e çıkartılması veri transferi verimliliğini arttırır. Jumbo frame uçtan uca yapılandırılmalıdır. Bir ESXi sunucu üzerinde Jumbo frame etkinleştirildiğinde fiziksel olarak bağlı olduğu anahtar bağlantısının MTU (maximum transmission unit) ile aynı miktar seçilmelidir.

Bir sanal makine üzerinde Jumbo frame etkinleştirilmesine iş yüküne bakarak karar verilmelidir. Eğer iş yükü tutarlı ve sürekli bir şekilde büyük miktarda ağ verisi aktarıyorsa Jumbo frame etkinleştirilebilir. Bu durumda sanal makine işletim sistemi ve NIC’i de Jumbo frame desteklemelidir.

Jumbo frame kullanmak vSphere vMotion performansını da artırır. Aktarım alanı trafiği taşıyan anahtarlar ve yönlendiriciler üzerinden VXLAN için en az 1600 byte’lık bir MTU boyu seçilmelidir.

Tablo 9 - Jumbo Frames Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-FİZ-AĞ-008 | Aşağıdaki tip trafiği taşıyan vDS bağlantı gruplarında ve fiziksel anahtar bağlantılarında en az 9000 byte (Jumbo frames) MTU boyu tanımlanmalıdır:   * NFS * vSAN * vMotion * VXLAN * vSphere Replication | MTU boyunu en az 9000 byte’a (Jumbo frames) e çıkartmak verimliliği artırır. VXLAN’ı destekleyebilmek için en az 1600 byte gerekirken tüm bağlantı grupları arasında tutarlı olabilmek adına o da 9000 byte varsayılan MTU boyu olarak seçilmelidir. | MTU paket boyunu ayarlarken, tüm ağ yolu boyunca bulunan cihazlar aynı MTU paket boyuna göre ayarlanmalıdır. |

* + 1. Fiziksel Veri Depolama Tasarım Kararları

Tasarımda çeşitli değişik veri depolama sistemleri kullanılabilir ve paylaşımlı veri depolama tasarımı bölümünde her birinin SDDC’deki yeri, kullanımı ve arka planlarına dair ayrıntılar verilecektir. Bu bölümde fiziksel veri depolama taraşımı anlatılmaktadır.

Bu tasarım VSAN kullanılacak şekilde yapılmakla birlikte, özellikle yönetim podları için herhangi desteklenen bir veri depolama sistemi de kullanılabilir.

* + - 1. VSAN Fiziksel Tasarımı

Yazılım tanımlı veri depolama SDDC için ana teknolojilerden birisidir. Bu tasarım yönetim kümelerinin yazılım tanımlı veri depolama ihtiyaçları için vSAN kullanmaktadır.

VSAN tamamıyla hipervizör entegre tümleşik veri depolama yazılımıdır. VSAN sunucu disklerinden ve SSD disklerinden bir küme oluşturarak sanallaştırma sunucularına ve sanal makinelere flash için optimize edilmiş yüksek dirençli paylaşımlı bir veri depolama sistemi olarak sunar. VSAN veri depolama politikaları aracılığıyla her bir sanal makine başına ayrı kapasite, performans ve erişilebilirliğin kontrol edilebildiği bir ortam sağlar.

* + - 1. Gereksinimler ve Bağımlılıklar

Yazılım tanımlı veri biriminin aşağıdaki gereksinimleri ve seçenekleri vardır:

* vSAN kümesine veri depolama kaynaklarını sunarak hizmet edecek en az 3 adet sanallaştırma sunucusu gereklidir.
* vSAN hibrid ya da tamamı ile flash veri depolama kaynakları ile yapılandırılırlar.
* Bir hibrid vSAN yapılandırması hem manyetik cihazlara hem de flash cache cihazlara ihtiyaç duyar.
* Tamamıyla flash bir yapılandırma için en az vSphere 6.0 sürümü gereklidir.
* Kümeye veri depolama kaynaklarını sunan tüm sanallaştırma sunucuları aşağıdaki tüm gereksinimleri sağlamalıdırlar:
  + En az bir SSD disk. SSD flash cache katmanı HDD kapasite katmanı kapasitesinin en az %10 kadar olmalıdır.
  + En az iki adet HDD.
  + RAID kontrolcüsü vSAN uyumlu olmalıdır.
  + vSAN trafiği için multicast etkinleştirilmiş en az 10GbE ağ genişliği.
  + vSphere HA izolasyon tepki ayarı sanal makineleri kapatmak şeklinde ayarlanmalıdır. Bu ayar ile izolasyon veya ağ bölümlenmesi gerçekleşirse ayrık beyin (split brain) durumu yaşama olasılığı ortadan kalkmaktadır. Ayrık beyin durumunda sanal makineler yanlışlıkla farklı iki sanallaştırma sunucusu tarafından birden ayağa kaldırılmaya çalışabilirler.

Tablo 10 - vSAN Fiziksel Veri Depolama Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-FİZ-DEP-001 | Yönetim podunda en az tek bir disk grubu yaratabilmek için bir ya da daha fazla 200GB veya daha büyük SSD ve iki ve daha fazla geleneksel 1TB ya da daha büyük HDD kullanılmalıdır. | Bir 200 GB SSD ve iki 1 TB HDD kullanarak yönetim sanal makineleri için gerekli kapasite ve %10 flash tabanlı cache sağlanmaktadır. | Sadece tek bir disk grubu kullanarak şerit miktarını (performansı) sınırlandırmış ve hata alanını büyütmüş olur. |

* + - 1. Hibrid ve Tamamıyla Flash Modları

vSAN hibrid ve tamamıyla flash olmak üzere iki ayrı operasyon modu sunmaktadır. Hibrid veri depolama mimarisinde, vSAN sunucu üzerinde bulunan kapasite cihazlarını (bu tasarımda manyetik cihazlar) ve tipik olarak SSD ya da PCI-e halindeki cache cihazlarını bir havuzda birleştirerek dağıtık paylaşımlı bir veri deposu olarak sunar. Tamamıyla flash veri depolamada da SSD veya PCI-e cihazlar sadece yazma önbelleği olarak kullanırken, diğer flash tabanlı cihazları ise kapasite katmanı için dayanıklılık ve veri kalıcılığı sağlar.

Tablo 11 - vSAN Modu Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-FİZ-DEP-002 | Yönetim kümesi içerisinde vSAN hibrid modda yapılandırılmalıdır. | Yönetim kümesi içindeki sanal makineler vSAN tarafından barındırılmakta ve yüksek performansa ihtiyaç duymamakta veya tamamıyla flash bir yapılandırmanın maliyetini gerektirmemektedir. | vSAN hibrid modu tamamıyla flash modunun sağladığı performansı ve tekilleştirme gibi yetenekleri verememektedir. |

* + - 1. Donanım Değerlendirmeleri

VMware vSAN küme donanımları kendi belirleyeceğimiz parçalarla oluşabildiği gibi vSAN hazır node listesinden de seçilebilir. Küme sanallaştırma sunucuları kendi belirlediğimiz bileşenlerden oluşacaksa seçilecek SSD, HDD bileşenleri VMware uyumluluk listesinde olmalı, I/O kontrolcüleri ise vSAN için onaylanmış sürücü ve mikro kod kombinasyonuna sahip olmalıdır. Bir vSAN hazır node ise OEM sunucu üreticisi ve VMware ile birlikte vSAN için test edilip onaylanmış bir hazır form faktörüdür. SDDC-FİZ-009 tasarım kararına dayanarak bu tasarımda vSAN hazır node kullanılacaktır.

* + - 1. SSD Karakteristikleri

Bir vSAN yapılandırılmasında SSD’ler hibrid modda önbellek olarak, tamamıyla flash modda ise hem önbellek hem de kapasite katmanı olarak kullanılmaktadır. Bir hibrid kurulumda SSD kullanımı yazma önbelleği (yaklaşık %30) ve okuma önbelleği olarak (yaklaşık %70) bölünmüştür. Sonuç olarak SSD’lerin dayanıklılığı ve birim zamanda sağlayabildiği I/O sayısı performansı önemli olarak etkilemektedir. Tamamıyla flash kurulumlarda dayanıklılık ve performans aynı kriterlere sahiptir. Bununla birlikte pek çok yazma operasyonu önbellek katmanında gerçekleşeceği için kapasite katmanının ömrünü uzatmaktadır.

* + - 1. SSD Dayanıklılığı

Bu tasarımda D sınıfı dayanıklılığa sahip SSD’ler önbellek katmanında kullanılmaktadır. vSAN için kullanılacak SSD’lerin dayanıklılığını belirlemek için sanayi standardı yazma metrikleri cihazların güvenilirliğini belirlemede ana etmen olarak kabul edilmektedir. Tüm üreticiler arasında ortak bir standart metrik olmamakla birlikte bir günde cihaza yazılan miktar (DWPD, drive writes per day) ya da yazılan petabyte (PBW, petabytes written) ölçümleri kullanılmaktadır.

vSphere 5.5 sürümünde dayanıklılık sınıfı DWPD’ye dayanmakta iken, vSAN 6.0 ve sonrası sürümlerde dayanıklılık sınıfı yazılan terabyte (TBW, terabytes written) kullanılmaya başlanmıştır. TBW kullanılmasının arkasındaki sebep nispeten düşük DWPD değerlerinde sahip ama yüksek kapasiteli cihazların kullanılabilmesi için esneklik sağlayabilmektir.

Bir üretici SSD’leri için ölçüm değeri olarak DWPD değerini sağlıyor ise TBW cinsinden dayanıklılık değeri aşağıdaki şekilde hesaplanabilir:

Önbellek ve kapasite katmanları için kullanılması planlanan SSD’ler için aşağıdaki tabloda vSAN hibrid ve tamamıyla flash modlarında kabul edilebilir dayanıklılık sınıfları gösterilmiştir:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Dayanıklılık Sınıfı | TBW | Hibrid Önbellek Katmanı | Tamamıyla Flash Önbellek Katmanı | Tamamıyla Flash Kapasite Katmanı |
| Class A | >=365 | Hayır | Hayır | Evet |
| Class B | >=1825 | Evet | Hayır | Evet |
| Class C | >=3650 | Evet | Evet | Evet |
| Class D | >=7300 | Evet | Evet | Evet |

Tablo 12 - SSD Dayanıklılık Sınıfı Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-FİZ-DEP-003 | Yönetim kümesi için önbellek katmanında Class D (>=7300) SSD’ler kullanılmalıdır. | Önbellek katmanına dahil bir SSD yıpranmadan dolayı arızalanırsa tüm disk grubu erişilmez hale gelir. Bu da potansiyel veri kaybına yol açabilir ya da operasyonel etkisi olur. | Yüksek dayanıklılık sınıflarına dahil olan SSD’lerin maliyetleri diğerlerine kıyasla daha yüksek olabilir. |

* + - 1. SSD Performansı

SSD’lerin performans sınıfı ile vSAN performansı seviyesi arasında direkt bir korelasyon vardır. En yüksek performansı gösteren donanımlar ile en iyi performansı gösteren çözüme ulaşılabilir. Dolayısıyla maliyet karar verici etmen olacaktır. Daha düşük sınıfa ait bir cihaz performans ya da kapasite olarak ideal değilken daha maliyet etkin olabilir. Optimal performans için E sınıfı ya da daha iyi SSD’ler seçilmelidir. Bir cihazın kapasitesi seçiminden önce disk grupları ve gelecekteki büyüme oranları da göz önünde bulundurulmalıdır. VMware performans sınıflarını[[19]](#footnote-19) aşağıdaki şekilde tanımlamıştır:

Tablo 13 - SSD Performans Sınıfları

|  |  |
| --- | --- |
| Performans Sınıfı | Saniyede Gerçekleşen Yazma Miktarı |
| Class A | 2500-5000 |
| Class B | 5000-10000 |
| Class C | 10000-20000 |
| Class D | 20000-30000 |
| Class E | 30000-100000 |
| Class F | 100000 ve üstü |

SSD kapasite boyu seçilirken kapasite katmanında tüketilecek miktarın en az %10 u tercih edilmelidir.

* + - 1. Önbellek Algoritması

Hem hibrid kümeler hem de tamamıyla flash kümeler kapasite katmanının %10 u boyutunda önbellek tavsiyesine uymalıdır. Bununla birlikte iki yapılandırma arasında farklar da vardır. Hibrid vSAN’da önbelleğin %70 i manyetik disklere erişimi azaltmak adına sıkça okunan disk bloklarını barındırır. Geriye kalan %30 ise yazmalar için ayrılır. Tamamıyla flash kümelerde ise iki tip flash bulunmaktadır: çok hızlı ve dayanıklı önbellek ve maliyet etkin kapasite flash katmanı. Burada önbelleğin %100 ü yazmalar için ayrılır çünkü okuma performansı yeterlidir.

Tablo 14 - SSD Performans Sınıfı Seçimi

|  |  |  |  |
| --- | --- | --- | --- |
| Tasarım Kalitesi | Seçenek 1 Class E | Seçenek 2 Class C | Yorumlar |
| Erişilebilirlik | Kaliteye etkisi yok. | Kaliteye etkisi yok. | Her iki seçenekte erişilebilirliği etkilemez. |
| Yönetilebilirlik | Kaliteye etkisi yok. | Kaliteye etkisi yok. | Her iki seçenekte yönetilebilirliği etkilemez. |
| Performans | Kaliteye pozitif etkisi var. | Kaliteye negatif etkisi var. | Depolama sınıfı ne kadar yüksek olursa performans o kadar yükselir. |
| Kurtarılabilirlik | Kaliteye etkisi yok. | Kaliteye etkisi yok. | Her iki seçenekte kurtarılabilirliği etkilemez. |
| Güvenlik | Kaliteye etkisi yok. | Kaliteye etkisi yok. | Her iki seçenekte güvenliği etkilemez. |

Tablo 15 - SSD Performans Sınıfı Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-FİZ-DEP-004 | Yönetim kümesi için Class E (saniyede 30000 ila 100000 yazma) tipi SSD’ler kullanılacaktır. | Yönetim kümesi için olan veri depolama I/O performans gereksinimleri en azından Class E tipi SSD kullanımını gerektirir. | Class E sınıfındaki SSD’lerin maliyetleri alt sınıflardaki SSD’lere kıyasla daha yüksek olabilir. |

* + - 1. Manyetik Hard Disk Drives (HDD) Karakteristikleri

Bir VMware vSAN ortamındaki HDD’lerin iki farklı amacı vardır. Manyetik diskler ya da HDD’ler önbellek katmanındaki SSD’lerden farklı olarak bir vSAN depolama ortamında kapasite katmanını oluştururlar. Bir sanal makinenin disk üzerindeki şerit genişliği aynı makinenin politika katmanında belirlenebilir. vSAN depolama politikaları dışında ek şerit genişlikleri kullanabilir ve farklı yerleştirme kararları alabilir. SAS ve NL-SAS (nearline SAS) en iyi sonuçları vermektedir. Bu tasarım maliyet ve erişilebilirlik arasındaki dengeyi korumak adına 10.000 RPM diskler kullanmaktadır.

* + - 1. HDD Kapasite, Maliyet, Erişilebilirlik Kavramları

VMware vSAN tasarımı kapasite katmanı için gereken manyetik disk adedini ve kapasite katmanının ne kadar etkin olacağını göz önünde bulundurmalıdır. SATA diskler genellikle daha yüksek kapasiteleri daha ucuz maliyetle sağlarken burada feda edilen performans olur. SATA disklerin performansı daha düşük dönme hızlarına (7200 RPM) sahip oldukları için SAS disklerin performansı kadar iyi değildir. Performansın kritik olduğu ortamlarda SAS manyetik diskler SATA disklere tercih edilmelidir. Yüksek kapasiteli bir diskin arızasının erişilebilirlik ve daha fazla bileşenin kurtarılmasına üzerinde operasyonel bir etkisi olacaktır.

* + - 1. Dönüş Hızı Kavramı

HDD’ler daha güvenilir olma eğilimindedir ama bununda bir maliyeti vardır. SAS diskler 15.000 RPM hızlarına kadar erişebilmektedir.

Tablo 16 - vSAN HDD Ortamı Karakteristikleri

|  |  |
| --- | --- |
| Karakteristik | Dakika Başına Dönme Hızı (RPM) |
| Kapasite | 7.200 |
| Performans | 10.000 |
| Ek Performans | 15.000 |

Önbellek kullanımına uygun iş yükleri disk performansı özelliklerine daha az duyarlıdırlar. Bununla birlikte iş yükleri zaman içerisinde değişiklik gösterebilirler. Kapasite katmanı seçilirken 10.000 RPM dönüş hızına sahip HDD’ler kabul edilmiş normlardır.

Yazılım tanımlı veri depolama modülü için VMware ortamın özelliklerine uygun bir HDD yapılandırması kullanılmasını tavsiye etmektedir. Belirli bir gereksinim yoktur ve 10.000 RPM disklerin seçilmesi maliyet ve erişilebilirlik arasında bir dengeye sahip olunmasını sağlar.

Tablo 17 - HDD Seçimi Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-FİZ-DEP-005 | Yönetim kümesi için 10.000 RPM dönüş hızında HDD’ler kullanılacaktır. | 10.000 RPM HDD’ler VMware vSAN için olan performans ve erişilebilirlik arasında bir denge sağlar.  10.000 RPM disklerin performansları disk boşaltma sorunlarına engel olur. Hibrid modda vSAN periyodik olarak önbellek de bekleyen yazma işlemlerini kapasite katmanına aktarır. | Daha yavaş ve daha ucuz HDD’ler burada kullanılamamaktadır. |

* + - 1. I/O Kontrol Kartları

vSAN yapılandırmalarında I/O kontrol kartı seçimi disk seçimi kadar önemlidir. vSAN pass-through ya da RAID 0 kipinde SAS, SATA ve SCSI kontrol kartlarını desteklemektedir. Bir sanallaştırma sunucu üzerinde birden fazla kontrol kartı olması da desteklenmektedir. Birden fazla kontrol kartı olması halinde performans iyileştirilebilir ve kart veya SSD arızaları daha küçük disk grupları ile sınırlandırılmış olur. Tek bir kontrol kartı durumunda tüm diskler aynı cihaz tarafından yönetildiğinden arıza durumunda tüm veri depolama etkilenir.

Kontrol kartının kuyruk derinliği kavramı performansı etkileyen en önemli bileşendir. VMware uyumlu donanımlar listesindeki tüm kontrol kartları en az 256 kuyruk derinliğine sahiptirler.

* + - 1. NFS Fiziksel Veri Depolama Tasarımı

NFS (Network File System) bir kullanıcının kendi bilgisayarından ağ üzerindeki dosyalara yerel veri depolama alanındaymış gibi erişebilmesine yarayan dağıtık bir dosya sistemi protokolüdür. Bu durumda son kullanıcı bilgisayarı ESXi sanallaştırma sunucusu ve veri depolama da NFS hizmetlerini sunan harici bir veri depolama sistemidir.

Yönetim kümeleri VMware vSAN’ı ana veri depolama kaynağı olarak kullanırken, NFS’i ikincil veri depolama olarak kullanabilirler. İşlem kümeleri herhangi bir veri depolama teknolojisi ile sınırlı değildirler. İşlem kümeleri için hangi teknolojinin kullanılacağına dair karar verirken göz önünde bulundurulması gereken kavramlar iş yükünün gerektirdiği performans, kapasite ve yeteneklerdir (replikasyon, tekilleştirme, sıkıştırma vb.).

Tablo 18 - NFS Kullanımı Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-FİZ-DEP-006 | NFS veri depolama sistemi yedekleme verisi, arşiv verisinin dışarı çıkartılacağı ve ISO ya da şablonların tutulacağı alanlar olarak yapılandırılarak kullanılacaktır. | Ana sanal makine depolama sistemini yedeklerin alınacağı veri depolama sisteminden ayırmak gereklidir. Ayrıca vRealize Log Insight yazılımı arşiv çıkabilmek için NFS’e ihtiyaç duyar. | NFS yeteneğine sahip harici bir depolama sistemi gereklidir. |

Tasarıma uygun bir NFS veri depolama sisteminin aşağıdaki gereksinimleri karşılaması gereklidir:

* Veri depolama sistemleri Leaf-Spine mimarideki ağa direkt olarak Leaf anahtarlara bağlanmalıdırlar.
* Tüm bağlantılar 10GbE Ethernet kullanılarak yapılmalıdır.
* Jumbo Frames etkinleştirilmelidir.
* Veri depolama sisteminde 10K SAS ya da daha hızlı diskler kullanılmalıdır.

Farklı hız ve kapasitede diskler kullanılarak veri depolama sistemi üzerinde farklı performans ve kapasite katmanları oluşturulabilir. Yönetim kümesi disk sistemi üreticisinin tavsiye ettiği RAID yapılandırmasında 10K SAS diskler kullanacaktır.

Tablo 19 - NFS Donanımı Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-FİZ-DEP-007 | NFS birimler için 10K SAS diskler kullanılacaktır. | 10K SAS diskler performans ve kapasite arasında dengeli bir seçimdir. Gerekirse daha hızlı diskler de kullanılabilir.  vSphere Data Protection yazılımı yedekleme ile ilgili SLA’leri karşılayabilmek için yüksek performansa ihtiyaç duymaktadır.  vRealize Automation yazılımı içerik kataloğu için yüksek performansa ihtiyaç duymaktadır.  vRealize Log Insight yazılımının arşiv verilerini uzun süre tutabilmek adına kapasite ihtiyacı vardır. | 10K SAS diskler alternatiflerinde göre daha maliyetli olabilirler. |

* + - 1. Birimler (Volumes)

Bir birim disk sistemi üzerinde birden çok diskin aynı RAID yapılandırmasına tabi olan koleksiyonuna verilen addır. Bir birim içerisinde birden çok veri deposu oluşturulabilir ancak yüksek I/O erişimi gerektirmeyen uygulamalar için bu yapı uygundur. Yüksek I/O gerektiren yedekleme gibi uygulamalar için atanmış tek bir birim ve tek bir veri deposu olmalıdır. Diğer aynı birim üzerinde barındırılan uygulamalar için depolama I/O kontrolü (SIOC) etkinleştirilerek yüksek I/O gerektiren uygulamaları sınırlandırılarak diğer uygulamaların da ihtiyaç duydukları kaynaklara erişebilmeleri garanti edilir.

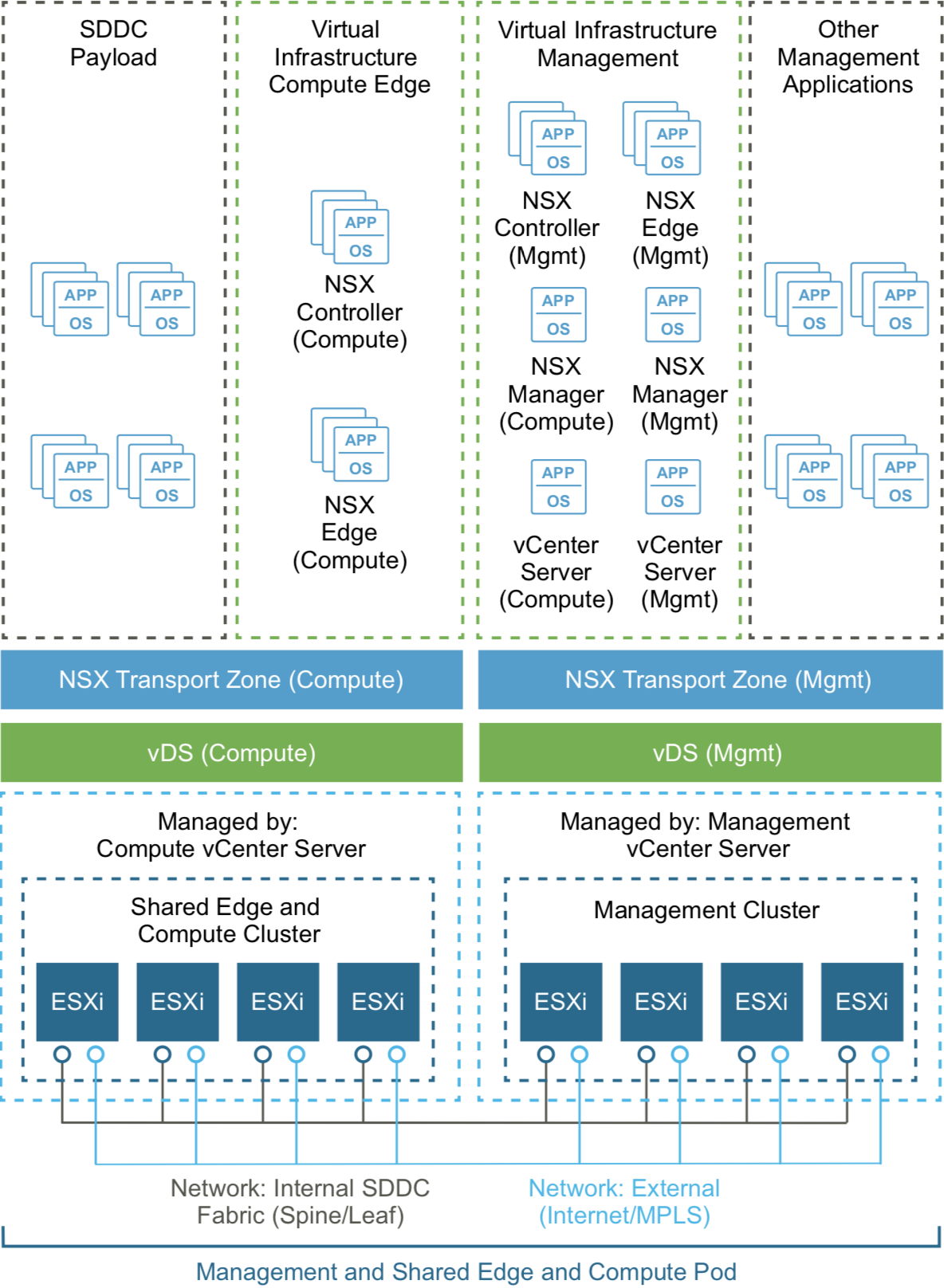
Tablo 20 - Birim Atama Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-FİZ-DEP-008 | Yedekleme gereksinimlerini sağlayabilmek için atanmış bir NFS Birimi kullanılacaktır. | Yedekleme ve geri dönüş işlemleri yüksek I/O gereksinimlerine sahiptir. Ayrı bir birim kullanılarak diğer yönetim birimlerini etkilemesinin önüne geçilmiş olunur. | Atanmış birimler veri depolama sistemi yöneticisi için ek iş yükü getirebilir. Ayrıca disk sistemin RAID tipine ya da cihaza göre atanmış birimler daha fazla disk adedi gerektirebilir. |
| SDDC-FİZ-DEP-009 | Diğer yönetim bileşenlerinin veri depoları için paylaşımlı bir birim kullanılacaktır. | Yönetim bileşenlerinin düşük I/O ya sahip yapıları sebebiyle paylaşımlı birimler kullanabilir. | Paylaşımlı birimler ve onları kullanan uygulamalar için yeterli miktarda kapasite ayrılmalıdır. |

* 1. Sanal Altyapı Tasarımı

Sanal altyapı tasarımı, altyapı katmanını oluşturan ve SDDC’nin iş sürekliliğini sağlayan yazılım bileşenlerinden oluşmaktadır. Bu bileşenler sanallaştırma platformu hipervizörünü, sanallaştırma yönetimini, veri depolama sanallaştırmasını, ağ sanallaştırmasını, yedekleme ve felaket kurtarma yazılım ürünlerinden oluşmaktadır. Bu katmandaki VMware ürünleri vSphere, vSAN, NSX, vSphere Data Protection ve VMware Site Recovery Manager’dir.

SDDC sanal altyapısı iki bölgeden oluşmaktadır. Her bölgede bir yönetim bir de paylaşımlı kenar ve işlem podu bulunmaktadır.



Yönetim podunda SDDC’yi yöneten sanal makineler çalışmaktaıd.r Bu sanal makineler, vCenter Server, NSX Manager, NSX Controller, vRealize Operations, vRealize Log Insıght, vRealize Automation, Site Recovery Manager ve diğer ortak yönetim bileşenleridir. Tüm yönetim, takip ve altyapı hizmetleri bu kritik hizmetlere yüksek erişilebilirlik sağlayan bir vSphere kümesine yerleştirilmiştir. Yönetim kümesindeki yetkiler erişimi sadece yöneticiler erişecek şekilde sınırlamaktadır. Bu yönetim, takip ve alt yapı sağlayan sanal makineleri korumak içindir.

Sanal alt yapı tasarımı aynı zamanda paylaşımlı bir kenar ve işlem podu da kullanmaktadır. Bu paylaşımlı pod tipik bir kenar ve işlem podunun özelliklerini tek bir podda birleştirmektedir. Eğer gerekirse gelecekte bu fonksiyonların ayrılması da mümkündür. Bu podun ana fonksiyonları aşağıdaki gibidir:

* Fiziksel ağa çıkan veya fiziksel ağdan gelen trafik bağlantılarını sağlamak.
* Fiziksel dünyadaki VLAN’lar ile bağlantı kurmak.
* SDDC misafir sanal makinelerini barındırmak.

Paylaşımlı kenar ve işlem podu NSX tarafından sağlanan sanal ağları (overlay networks) harici ağlarla birleştirir. Bir SDDC’de farklı tipteki işlem podları karıştırılarak farklı SLA’ler için farklı işlem havuzları oluşturulabilir.

* + 1. ESXi Tasarımı

ESXi tasarımında sanallaştırma sunucularının başlatma seçenekleri (boot), kullanıcı erişimi ve sanal makinelerin swap yapılandırmaları bulunmaktadır.

* + - 1. ESXi Donanım Gereksinimleri

ESXi donanım gereksinimleri fiziksel tasarım temelleri başlığı altında sunulmuştur. Aşağıdaki tasarımda ESXi yapılandırmasına dair bilgiler verilmektedir.

* + - 1. ESXi Elle Kurulum ve Başlatma Seçenekleri

ESXi 6.5 aşağıdaki veri depolama sistemlerine kurulabilir ve onlardan başlayabilir:

* SATA Disk sürücüleri
* Serial Attached SCSI (SAS) Disk sürücüleri
* SAN
* USB Cihazlar
* FCoE (Yazılımsal Fibre Channel over Ethernet)

ESXi sistemin mikro kodu desteklediği sürece 2TB dan büyük disklerden başlatılabilir.

* + - 1. ESXi Başlatma Diski ve Yaz Boz Alanı (Scratch) Yapılandırması

ESXi’ın yeni kurulumlarında yazılım bir 4GB’lık VFAT yaz boz alanı oluşturmaktadır. ESXi bu alanı log dosyalarını sabit olarak saklamak için kullanmaktadır. Varsayılan ayarlarla sorun gidermek için kullanılan vm-support çıktıları da yaz boz alanında saklanmaktadır.

USB ortamı üzerine yapılan ESXi kurulumlarında varsayılan yaz boz alanı oluşturulmaz. VMware yaz boz alanı olarak paylaşımlı bir veri depolama üzerinde oluşturulacak bir alanın tahsis edilmesini ve uzak syslog sunucusuna log dosyalarının gönderilecek şekilde yapılandırma yapılmasını tavsiye etmektedir.

Tablo 21 - ESXi Başlatma Diski Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-ESXi-001 | Tüm ESXi sanallaştırma sunucuları en az 16GB lık bir SD cihaza kurulup oradan başlatılacak şekilde yapılandırılmalıdır. | SD kartları ESXi kurulumu için kullanılabilecek ucuz ve basit bir yöntemdir. SD Kart kullanarak tüm yerel HDD’lerin vSAN dahilinde kullanılması mümkündür. | SD kart kullanıldığında log dosyaları yerel olarak tutulmamaktadır. |

* + - 1. ESXi Sunucuya Erişimler

Kurulumdan sonra ESXi sanallaştırma sunucuları bir VMware vCenter Server sistemine dahil edilerek oradan yönetilmeye başlarlar. Sanallaştırma sunucusu konsoluna direkt erişim hala mümkün olmakta ve bu yöntem genellikle sorun gidermek için kullanılabilmektedir. Bir ESXi sanallaştırma sunucusuna aşağıdaki üç yöntem ile erişmek mümkündür:

* Direkt Konsol Kullanıcı Ara Yüzü (DCUI). Konsolun grafik tabanlı ara yüzüdür. Temel idari işlerin gerçekleştirilmesine ve sorun gidermek için kullanılabilir.
* ESXi Shell. ESXi konsolunun Linux bash tipi komut satırıdır.
* Secure Shell (SSH) Erişimi. Uzaktan komut satırı bağlantısıdır.

Her yöntem etkinleştirilip, devre dışı bırakılabilinir. Var sayılan ayarlarda ESXi Shell ve SSH kipleri ESXi sanallaştırma sunucusunun güvenliği için devre dışıdır. DCUI ise sadece “Strict Lockdown” kipinde devre dışı bırakılabilinir.

* + - 1. ESXi Kullanıcı Erişimi

Var Sayılan ayarlarla “root” kullanıcısı bir ESXi sanallaştırma sunucusuna direk olarak bağlanabilen tek kullanıcıdır. Bununla birlikte ESXi sanallaştırma sunucusu bir Active Directory etki alanına eklenirse erişimler Active Directory grupları vasıtasıyla da yapılabilir. Sanallaştırma sunucusuna kimin bağlandığının denetimini yapmak ise kolaylaşır.

Tablo 22 - ESXi Kullanıcı Erişimi Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-ESXi-002 | Her bir sanallaştırma sunucusu içinde bulunduğu bölgenin Active Directory etki alanına dahil edilecektir. | Active Directory üyeliği kullanımı ESXi sanallaştırma sunucusuna erişim sağlarken esneklik sağlar. Her kullanıcının özgün bir hesap ile bağlanması denetim açısından daha iyi bir görünürlük sağlar. | Sanallaştırma sunucularının Active Directory’e eklenmesi yönetimsel ek bir yük getirebilir. |
| SDDC-SA-ESXi-003 | Var sayılan ESX Admins grubu SDDC-Admins Active Directory grubu ile değiştirilecektir. ESXi yöneticileri SDDC-Admins grubuna eklenecektir | SDDC-Admins grubunun olması bilinen bir yönetimsel erişim noktasını değiştirdiği için daha güvenlidir. Ayrıca farklı grupların varlığı yönetimsel işlerin ayrılabilmesine olanak verir. | Sanallaştırma sunucusunun gelişmiş ayarlarında değişiklik yapılması gereklidir. |

* + - 1. Sanal Makine Swap Yapılandırması

Bir sanal makine başlatıldığında sistem, sanal makinenin hafızasını desteklemek için bir VMkernel swap dosyası oluşturur. Var sayılan swap dosyası sanal makinenin yapılandırma dosyaları ile aynı yerde oluşturulur. Bu yapılandırmayı kolaylaştırırken, gereksiz bir replikasyon trafiği oluşturur.

Swap dosyasının yerini kullanıcının belirlediği bir yer ile değiştirerek replike olan trafik miktarı azaltılabilir. Bununla birlikte swap dosyasının yeniden oluşturulması gereken durumlarda vMotion operasyonu daha uzun sürebilir.

Tablo 23 - Diğer ESXi Sunucusu Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-ESXi-004 | Tüm ESXi sanallaştırma sunucuları merkezi NTP sunucusu ile senkronize olacaktır. | NTP sunucusu kullanılmadığında bir ESXi sanallaştırma sunucusu üzerine vCenter Server Appliance konumlandırıldığında başarısız olabilir. | Tüm ESXi sanallaştırma sunucusu ile NTP sunucusu arasındaki tüm güvenlik duvarları ilgili trafiğe izin vermelidir. |

* + 1. vCenter Server Tasarımı

vCenter Server tasarımı hem vCenter Server’ın kendisi ile ilgili tasarımı hem de VMware Platform Services Controller’a dair tasarım kararlarını içermektedir.

Bir Platform Services Controller, vCenter Single Sign-On, Licence Service, Lookup Service ve VMware Certificate Authority’yide içeren bir seri alt yapı hizmetini gruplayarak bir arada sunar. Platform Service Controller ile bağdaştırılmış vCenter Server aynı sanal makinede barınabileceği gibi ayrı sanal makineler halinde de (harici Platform Services Controller) yapılandırılabilir.

* + - 1. vCenter Server Konumlandırma

vCenter Server konumlandırma tasarım kararları kaç adet vCenter Server ve Platform Services Controller olacağı, kurulum tipi ve topoloji hakkındaki kararlardır.

Tablo 24 - vCenter Server Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-VC-001 | Her bölgenin ilk erişilebilirlik alanında iki adet vCenter Server kurulacaktır.  Bir vCenter Server SDDC yönetim bileşenlerini yönetecek, diğer vCenter ise kenar bileşenleri ve iş yükü bileşenlerini yönetecektir. | vCenter arızaları yönetim ya da iş yükleri için izole edilmiş olur. vCenter operasyonları yönetim ve işlemler için ayrılmış olur. Ölçeklenebilir bir küme yönetimini destekler.  İş yükleri için kapasite planlaması yapılırken vCenter ayrı tutulacağından yönetimsel kapasite bu plandan ayrı tutularak planlama kolaylaştırılır.  vSphere ortamlarının ve ilgili bileşenlerinin güncellenmeleri için bağımsız bakım zamanları oluşturulabilir. Bu bakımlar esnasında yönetim kısmı güncellenirken iş yükü etkilenmez, iş yükü kısmı güncellenirken yönetim kısmı işlevselliğine devam edebilir. | Her vCenter Server için ayrı bir lisans gerekmektedir. |

vCenter Server bir Windows tabanlı sistem olarak ya da Linux tabanlı VMware vCenter Server Appliance olarak kurulabilir. Linux tabanlı vCenter Server önceden yapılandırılmıştır ve hızlı kurulum sağlamakla birlikte gerekli Microsoft lisanslarından da tasarruf sağlar.

Tablo 25 - vCenter Server Platform Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-VC-002 | Tüm vCenter Server örnekleri Linux tabanlı vCenter Server Appliance olarak kurulacaktır. | Hızlı kurulum, ölçeklenebilirliği artırır ve Microsoft lisanslarından tasarruf etmeyi sağlar. | Linux tabanlı sanal makinelerde oluşabilecek sorunların giderilmesi için operasyon ekibinin Linux deneyimi olması gerekir. |

* + - 1. Platform Hizmetleri Kontrolcüsü Tasarım Genel Bilgiler

vCenter Server gömülü bir platform hizmetleri kontrolcüsü ile ya da harici olarak kurulumu desteklemektedir.

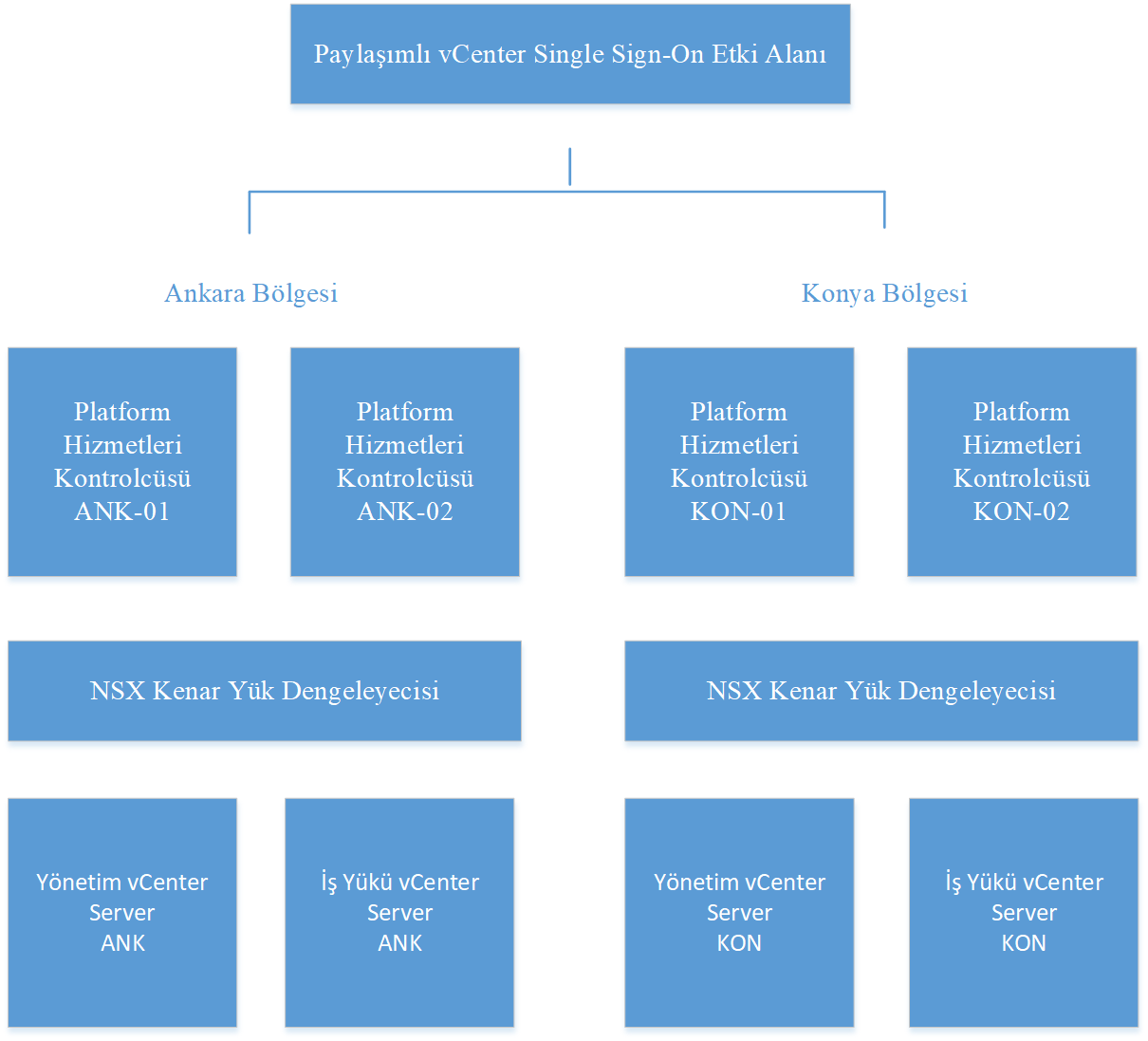
Platform hizmetleri kontrolcüsünün gömülü olduğu kurulumlarda vCenter Server ve platform hizmetleri kontrolcüsü aynı sanal makine içerisinde çalışır. Gömülü kurulumlar, tek bir vCenter Server’ın olduğu bağımsız ortamlarda tavsiye edilmektedir.

Harici platform hizmetleri kontrolcüsüne sahip ortamlarda birden fazla vCenter Server sistemi barınabilmektedir. vCenter Server sistemleri aynı platform hizmetleri kontrolcüsünü kullanabilmektedir. Örnek vermek gerekirse birçok vCenter Server kimlik doğrulama için aynı vCenter Single Sign-On hizmetini kullanabilmektedirler.

Eğer diğer platform hizmetleri kontrolcüleri ile eşleşme (replikasyon) yapma ihtiyacı varsa ya da çözüm bir ya da birden fazla vCenter Single Sign-On hizmeti içeriyorsa birden fazla harici platform hizmetleri kontrolcüsü farklı sanal makineler üzerinde konumlandırılabilir.

Tablo 26 - Platform Hizmetleri Kontrolcüsü Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-VC-003 | Tüm vCenter Serverları harici platform hizmetleri kontrolcüsü ile kurulmalıdır. | Platform hizmetleri kontrolcüleri arasında eşleşme yapabilmek için Harici platform hizmetleri kontrolcüleri gereklidir. | Yönetilmesi gereken sanal makine sayısı artar. |
| SDDC-SA-VC-004 | Tüm platform hizmetleri kontrolcüleri tek bir vCenter Single Sign-On etki alanına bağlanmalıdır. | Tüm platform hizmetleri kontrolcüleri tek bir vCenter Single Sign-On etki alanında birleştikleri zaman kimlik doğrulama ve lisans verilerini tüm bileşenler ve tüm bölgeler arasında paylaşabilirler. | Tek bir Single Sign-On etki alanı olacaktır. |
| SDDC-SA-VC-005 | Platform hizmetleri kontrolcüleri için ring topolojisi oluşturulacaktır. | Var sayılan ayarlar ile platform hizmetleri kontrolcüleri sadece bir başka kontrolcü ile eşleşebilir. Bu da eşleşme için tek bir noktadan kaynaklanabilecek hata olasılığı vardır. Ring topolojisinde her bir kontrolcünün iki eşleşme ortağı olacağından bu hata noktasını elimine eder. | Ring topolojisi yapılandırması için komut satırından girilecek gelişmiş komutlar gereklidir. |
| SDDC-SA-VC-006 | Platform hizmetleri kontrolcüleri için NSX Edge Services Gateway yazılımını yük dengeleyici olarak kullanılacaktır. | Bir yük dengeleyici kullanarak uygulamaların tüm PSC’lere erişilebilirliği artmış olur. | Yük dengeleyici yapılandırılması ve vCenter’ların yük dengeleyicinin sanal IP’sine yönlendirilmesi fazladan yönetimsel iş yükü getirir. |



Şekil 1 - vCenter Server ve Platform Hizmetleri Kontrolcüsü Kurulum Modeli

* + - 1. vCenter Server Ağı

Fiziksel ağ tasarımı bölümünde belirtildiği gibi, tüm vCenter Server sistemleri statik IP adresleri ve sunucu adları kullanmalıdır. IP adreslerinin dahili DNS sisteminde geçerli düz ve ters çözümlenebilecek kayıtları bulunmalıdır.

vCenter Server sistemlerinin aşağıdaki tüm bileşenler ile sürekli ağ bağlantıları sağlaması gerekmektedir:

* Tüm VMware vSphere istemciler vSphere Web Client kullanıcı ara yüzleri ile
* vCenter Server eklenebilir modül çalıştıran tüm sistemler ile
* Her bir ESXi sanallaştırma sunucusu ile
  + - 1. vCenter Server Yedekliliği

Merkezi yönetim noktası ve SDDC’nin takibini yapan yazılım olduğundan vCenter Server sistemlerinin korunması önemlidir. vCenter Server’ın nasıl korunacağı ne kadar kesintinin kabul edilebileceğine ve yük devretme mekanizmasının otomasyonunun gerekliliğine bağlıdır.

Aşağıdaki tabloda vCenter Server sistemlerinin koruma yöntemleri gösterilmiştir:

Tablo 27 - vCenter Server Sistemi ve vCenter Appliance Koruma Yöntemleri

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Yedeklilik Yöntemi | vCenter Server (Windows) Koruması | PSC (Windows) Koruması | vCenter Server (Appliance) Koruması | PSC (Appliance) Koruması |
| vSphere HA Kullanarak Otomatik Koruma | Evet | Evet | Evet | Evet |
| Elle yapılandırma ve elle yük devretme (cold standby) | Evet | Evet | Evet | Evet |
| Harici yük dengeleyicili HA küme | Mümkün değil. | Evet | Mümkün değil. | Evet |
| vCenter Server HA | Mümkün değil. | Mümkün değil. | Evet | Mümkün değil. |

Tablo 28 - vCenter Server Koruma Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-VC-007 | Tüm vCenter Server ve Platform Hizmetleri Kontrolcüleri vSphere HA kullanılarak korunacaktır. | Bir kesinti durumunda elle müdahale gerektirmeden vCenter Server’lar için erişilebilirlik hedeflerini sağlamaktadır. | vSphere HA yük devretme esnasında vCenter erişilemez durumda olacaktır. |

* + - 1. vCenter Server Appliance Boyutlandırması

Aşağıda verilen tablolarda yönetim vCenter Server ve iş yükü vCenter Server için gereken en az donanım gereksinimleri gösterilmektedir:

Tablo 29 - Yönetim vCenter Server'ı için Mantıksal Spesifikasyonlar

|  |  |
| --- | --- |
| Öznitelik | Spesifikasyon |
| vCenter Server sürümü | 6.5 (vCenter Server Appliance) |
| Fiziksel ya da Sanal sistem | Sanal (Appliance) |
| Appliance Boyutu | Küçük (100 sanallaştırma sunucusu ya da 1000 sanal makineye kadar) |
| Platform Hizmetleri Kontrolcüsü | Harici |
| CPU sayısı | 4 |
| Hafıza | 16 GB |
| Disk Alanı | 300 GB |

Tablo 30 – İş Yükü vCenter Server'ı için Mantıksal Spesifikasyonlar

|  |  |
| --- | --- |
| Öznitelik | Değer / Ayar |
| vCenter Server sürümü | 6.5 (vCenter Server Appliance) |
| Fiziksel ya da Sanal sistem | Sanal (Appliance) |
| Appliance Boyutu | Büyük (1000 sanallaştırma sunucusu ya da 10000 sanal makineye kadar) |
| Platform Hizmetleri Kontrolcüsü | Harici |
| CPU sayısı | 16 |
| Hafıza | 32 GB |
| Disk Alanı | 650 GB |

Tablo 31 - vCenter Server Appliance Boyutu Tasarım Kararları

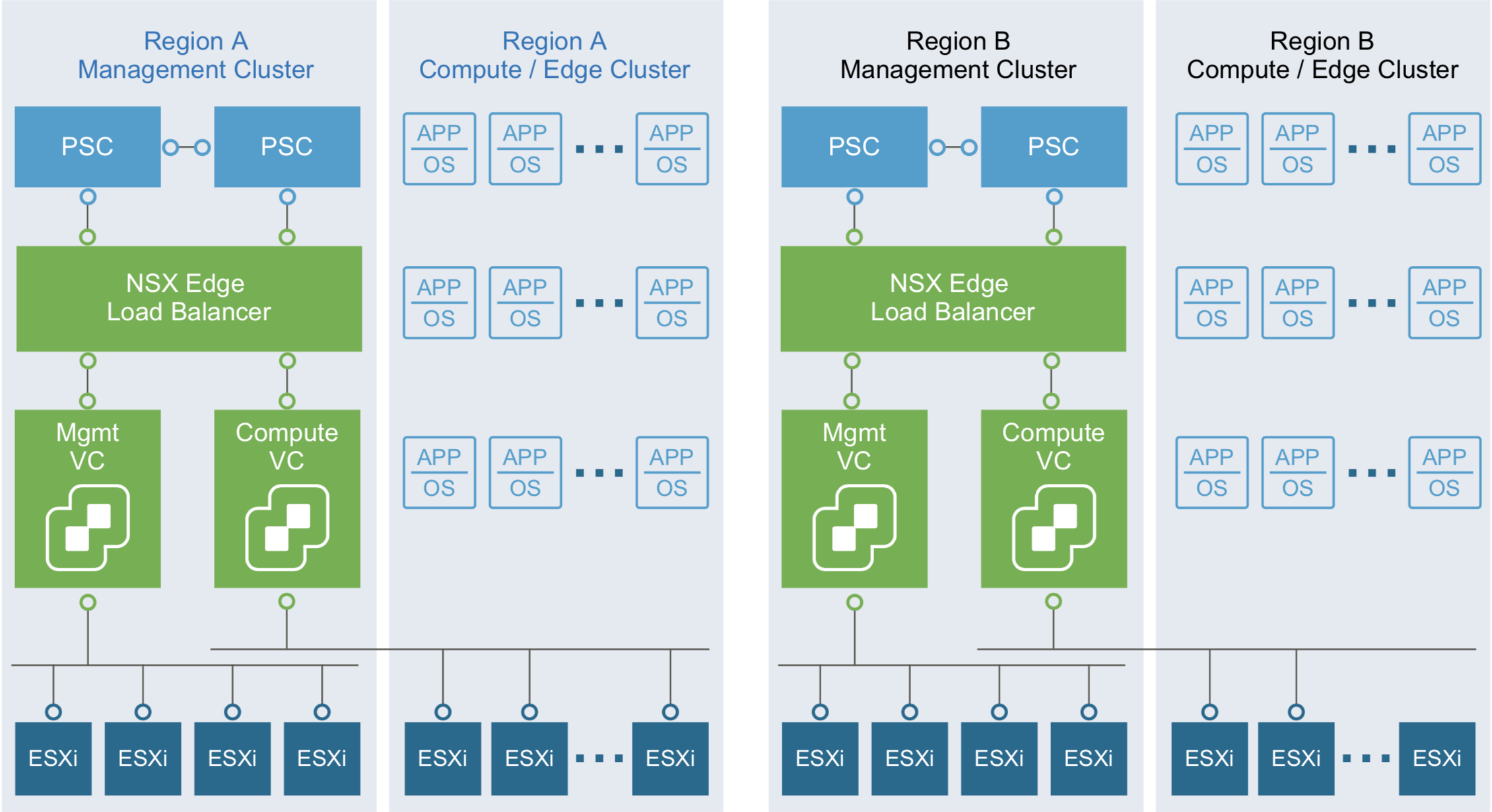
|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-VC-008 | Yönetim vCenter Server’ı en az küçük ebatlı olarak yapılandırılmalıdır. | Çalışacak yönetim sanal makinelerinin sayısına bakılarak vCenter Server’ın küçük boyutu ile yapılandırılması yeterlidir. | Yönetim ortamındaki sanal makinelerin sayısı değişirse vCenter Server’ın boyutu değiştirilmek zorunda kalınabilir. |
| SDDC-SA-VC-009 | İş yükü vCenter Server’ı en az büyük ebatlı olarak yapılandırılmalıdır. | Çalışacak olan iş yüklerine ve NSX kenar cihazlarının sayısına bakılarak vCenter Server’ın büyük boyutu ile yapılandırılması tavsiye edilmektedir. | İş yükü ortamı genişledikçe vCenter Server’ı X-Geniş boyutuna çekmek zorunluluğu ortaya çıkabilir. |

* + - 1. vSphere Küme Tasarımı

Küme tasarımı yapılırken içerisinde çalışacak iş yükünü hesaba katmak gerekmektedir. Bu tasarımdaki farklı küme tipleri farklı özelliklere sahiptirler.

Aşağıdaki maddeler küme tasarımı yaparken göz önünde bulundurulması ve hakkında karar verilmesi gereken buluşsal kavramlardır:

* Daha az sayıda büyük sanallaştırma sunucuları mı kullanılacak yoksa çok sayıda küçük sanallaştırma sunucuları mı kullanılacak? Dikey büyümesi öngörülen kümelerde az sayıda büyük sanallaştırma sunucuları bulunurken, yatay büyümesi öngörülen kümelerde çok sayıda küçük sanallaştırma sunucuları bulunur. Sanallaştırılmış bir sunucu kümesinde genellikle daha çok sayıda sanallaştırma sunucusu ve her biri üzerinde az sayıda sanal makine bulunur.
* Az sayıda büyük sanallaştırma sunucusu ile çok sayıda küçük sanallaştırma sunucusu almanın ilk alım maliyeti (capital cost) kıyaslanmalıdır. Maliyetler üreticiden üreticiye ve modelden modele farklılık gösterebilir.
* Az sayıda ya da çok sayıda sanallaştırma sunucusu yönetmenin operasyonel maliyetini değerlendirmek gereklidir.
* Kümenin amacını düşünmek gerekir.
* Toplam sanallaştırma sunucusu adedini ve küme limitlerini değerlendirmek gerekir.



Şekil 2 - Mantıksal vSphere Küme Yerleşimi (!!! tekrar çiz)

* + - 1. vSphere Yüksek Erişilebilirlik Tasarımı

VMware vSphere yüksek erişilebilirlik (vSphere HA) bir sanallaştırma sunucusu arızası durumunda arızadan etkilenen sanal makineleri küme içerisindeki başka sanallaştırma sunucusu üzerinde yeniden başlatarak sanal makineleri korur.

Kümenin yapılandırılması esnasında, sanallaştırma sunucuları kendi aralarında bir başkan (master) sanallaştırma sunucusu seçerler. Başkan sunucu vCenter Server sistemi ile iletişimde bulunur ve küme içerisindeki sanal makineleri ve diğer sanallaştırma sunucularını takip eder.

Başka sunucu farklı tipte arızaları takip eder:

* Beklenmedik bir enerji kesintisi gibi sanallaştırma sunucusu arızalarını
* Sanallaştırma sunucularının ağdan izole olma durumlarını veya bağlantı kesintilerini
* Veri depolama ile olan kesintileri
* Sanal makinelerin işletim sistemleri ile ilgili problemleri

Tablo 32 - vSphere HA Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-VC-010 | Tüm kümeleri arızlara karşı korumak için vSphere HA kullanılacaktır | vSphere HA hem sanal makineler için hem de sanallaştırma sunucuları erişilebilirliği için dirençli bir seviyede koruma sağlar. | Geriye kalan sanallaştırma sunucuları üzerinde göç eden ya da yeniden başlatılan sanal makineler için yeterli miktarda kaynak ayrılmış olmalıdır. |
| SDDC-SA-VC-011 | vSphere HA Sanallaştırma sunucusu izolasyon müdahale ayarı sanal makinelerin kapatılması olarak ayarlanacaktır. | vSAN HA izolasyon müdahale ayarının “power off” olması ve sanal makinelerin diğer sanallaştırma sunucularından yeniden başlatılmasını gerektirir. | Yanlış pozitif bir alarm durumunda sanal makineler kapatılacak ve sanallaştırma sunucusu izole oldu şeklinde işaretlenecektir. |

vSphere HA kabul kontrol politikası bir sistem yöneticisinin kümenin mevcut kaynaklarını nasıl yapılandıracağını belirler. Küçük bir vSphere HA kümesinde, kaynakların büyük bir kısmı sanallaştırma sunucusu arızası durumunda, politika uyarınca kullanılmak üzere rezerve tutulur.

Küme için seçilebilecek aşağıdaki politikalar mevcuttur:

* Kümenin kaldırabileceği sanallaştırma sunucusu arıza sayısı. vSphere HA belirli sayıda sanallaştırma sunucusunu devre dışı kalsa dahi kalanların gerekli miktarda kaynağa sahip olacağını garanti eder.
* Rezerve küme kaynakları yüzdesi. Toplam küme kaynaklarının yüzde cinsinden ne kadarının arıza durumlarında kullanılacağının belirlenmesidir. vSphere HA CPU ve hafızanın belirlenen yüzdesini arıza halinde sanal makinelerin kullanımı için rezerve tutar.
* Yük devretme sanallaştırma sunucuları. Bir sanallaştırma sunucusu arızalandığında vSphere HA etkilenen sanal makineleri belirli sanallaştırma sunucuları üzerinde yeniden başlatmaya çalışır. Eğer yeniden başlatmak mümkün olmazsa, mesela belirlenen sanallaştırma sunucuları üzerinde yeterli kaynak yoksa ancak o zaman kümenin kalan sunucuları üzerinde yeniden başlatmayı dener.
  + - 1. vSphere Küme İş Yükü Tasarımı

Bu tasarım aşağıda verilen şekilde vSphere kümeleri ve sorumlu oldukları iş yüklerini tanımlar:

Tablo 33 - vSphere Küme İş Yükü Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-VC-012 | Tüm yönetim sanallaştırma sunucularını içeren bir sanallaştırma kümesi oluşturulacaktır. | İşlem iş yüklerini yönetim iş yüklerinden ayırarak yapılandırma basitleştirilir. İşlem iş yüklerinin yönetim üzerinde etkisi olmaması sağlanır. Gerekirse kümeye yeni ESXi sanallaştırma sunucuları eklenebilir. | Birden çok kümelerin ve vCenter’ların yönetimi operasyonel iş yükünü artırır. |
| SDDC-SA-VC-013 | İşlem iş yüklerini, NSX kontrolcülerini ve ilgili NSX kenar cihazlarını barındıracak paylaşımlı kenar ve işlem kümesi oluşturulacaktır. | Yapılandırmayı basitleştirir ve ilk kurulum için gereken sanallaştırma sunucusu sayısını azaltır. Yönetim iş yüklerinin işlem iş yükleri üzerinde etkisi olmaması sağlanır. Gerekirse kümeye yeni ESXi sanallaştırma sunucuları eklenebilir. | Birden çok kümelerin ve vCenter’ların yönetimi operasyonel iş yükünü artırır. Kümenin paylaşımlı yapısından dolayı yeni iş yükleri eklendikçe kümenin ağ performansını sağlayacak şekilde ölçeklendirilmesi gerekir.  Paylaşımlı yapıdan dolayı kaynak havuzları oluşturulması gerekir. |

* + - 1. Yönetim Kümesi Tasarımı

Yönetim kümesi tasarımı sanallaştırma sunucu sayısını ve küme için olan vSphere HA ayarlarını belirler.

Tablo 34 - Yönetim Kümesi Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-VC-014 | Yönetim kümesi için 4 Sanallaştırma sunucusu kullanılacaktır | vSAN kümesi için gereken n+1 yedeklilik 3 sanallaştırma sunucusu ile sağlanmaktadır. Dördüncü sunucu ise bakım ve arıza durumlarında vSAN için n+1 yedekliliği saplamaktadır. | Yedekliliği saplamak için ek sanallaştırma sunucusu kaynakları gerekir. |
| SDDC-SA-VC-015 | Kabul kontrol politikası olarak 1 sanallaştırma sunucusu arızası ve yüzde cinsinden kaynak rezervasyonu ayarlanacaktır. | Yüzde cinsinden rezervasyon kullanmak özellikle sanal makinelerin değişken ve kayda değer miktarda işlemci ve hafıza rezervasyonları olduğunda oldukça işe yaramaktadır. | Dört sanallaştırma sunucusuna sahip yönetim kümesinde sadece üç sunucunun kaynakları kullanıma müsaittir. |
| SDDC-SA-VC-016 | Yönetim kümesi için sanallaştırma sunucusu profili oluşturulacaktır. | Sanallaştırma sunucusu profili kullanmak yapılandırmayı basitleştirdiği gibi ayarların tüm küme içerisinde tutarlı olmasını sağlar. | Bir sanallaştırma sunucusu üzerinde bir değişiklik yapıldığı zaman bu değişikliğin profile de yansıtılması gereklidir yoksa o sunucu profile uyumsuz duruma düşecektir |

Aşağıdaki tabloda yönetim kümesinin mantıksal tasarımına ait özellik ve ayarlar özetlenmiştir:

Tablo 35 - Yönetim Kümesi Mantıksal Tasarım Değerleri

|  |  |
| --- | --- |
| Özellik | Değer / Ayar |
| Yönetim sunucularını aşırı yüklenmeden destekleyebilecek sanallaştırma sunucu sayısı | 2 |
| Operasyonel kısıtlamalar (HA’den taviz vermeden bir sanallaştırma sunucusunu devre dışı bırakabilme) göz önünde bulundurularak tavsiye edilen sanallaştırma sunucu sayısı | 3 |
| vSAN kullanılırken, operasyonel kısıtlamalar (HA’den taviz vermeden bir sanallaştırma sunucusunu devre dışı bırakabilme) göz önünde bulundurularak tavsiye edilen sanallaştırma sunucu sayısı | 4 |
| Küme başına sanallaştırma sunucusu arıza kapasitesi | %25 rezerve CPU ve RAM |

* + - 1. Paylaşımlı Kenar ve İşlem Kümesi Tasarımı

Üzerinde misafir iş yükleri çalışan ESXi sanallaştırma sunucuları paylaşımlı kenar ve işlem kümelerinde yer alırlar. Paylaşımlı doğalarından dolayı NSX kontrolcüleri ve kenar cihazlar da bu kümede çalışırlar. Tasarım kararları ile sanallaştırma sunucularının sayısı, vSphere HA ayarları ve diğer küme karakteristikleri belirlenmektedir.

Tablo 36 - Paylaşımlı Kenar ve İşlem Kümesi Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-VC-017 | NSX Kontrolcüsü ve NSX kenar geçit cihazları için paylaşımlı kenar ve işlem kümesi oluşturulacaktır. | NSX Manager yazılımı vCenter Server sistemi ile bire bir ilişki gerektirmektedir. | Her bir yeni işlem vCenter sistemi kurulduğunda yeni bir de NSX Manager yazılımı kurulması gerekir. Her kontrolcüyü ayrı sanallaştırma sunucusu üzerinde tutmak için kural tanımlanmalıdır. |
| SDDC-SA-VC-018 | Kabul kontrol değerleri bir sanallaştırma sunucu kaybını karşılayacak ve yüzde cinsinden yük devretme kapasitesi olarak ayarlanacaktır. | vSphere HA, NSX kontrolcülerini ve geçit cihazlarını sunucu arızalarına karşı korumaktadır. HA sanal makineleri ayakta olan diğer sunucularda yeniden başlatır. | Potansiyel kaynak sıkıntısı olmadan önce tek bir sanallaştırma sunucusu kaybı kaldırılabilir. |
| SDDC-SA-VC-019 | En az dört sanallaştırma sunucusundan oluşan bir paylaşımlı kenar ve işlem kümesi oluşturulacaktır. | Yeterli yedeklilik ve çoğunluk oylaması için 3 adet NSX kontrolcüsü gereklidir. Bir yedek sanallaştırma sunucusu planlı bakım ve arıza durumlarında yük devralabilmek için yeterlidir. | Paylaşımlı kenar ve işlem kümeleri için 4 sanallaştırma sunucusu en küçük kabul edilebilir adettir. Dolayısıyla 3 sunucuya kıyasla bir miktar maliyet artışı olacaktır. |
| SDDC-SA-VC-020 | Paylaşımlı kenar ve işlem kümesi sunucularının harici erişimleri için VLAN destekli port grupları oluşturulacaktır. | Kenar geçitleri yönetim ağına olduğu gibi harici ağlara da erişim sağlamalıdır. | VLAN destekli port grupları doğru sayıda portlar ile ya da esnek port tahsisi ile yapılandırılmalıdır. |
| SDDC-SA-VC-021 | NSX kontrolcüleri ve kenar cihazlar için bir kaynak havuzu oluşturulacak, işlemci hissesi yüksek ve 16GB hafıza rezervasyonu ile yapılandırılacaktır. | NSX bileşenleri SDDC’ye giren ve çıkan tüm trafiği kontrol eder ve SDDC için trafik için yönlendirme bilgilerini günceller. Kaynak sıkıntısı durumunda bu sanal makinelerin gerekli kaynaklara erişmesi kritiktir. | Kaynak sıkıntısı durumunda NSX bileşenleri diğer iş yüklerine kıyasla daha fazla kaynak alacağından takip ve kapasite yönetimi işleri ön etkin olarak gerçekleştirilmelidir. |
| SDDC-SA-VC-022 | Tüm kullanıcı NSX kenar cihazları için oluşturulacak kaynak havuzu normal işlemci ve normal hafıza hisseleri ile yapılandırılacaktır. | Kullanıcılar için olan NSX kenar cihazları vRealize Automation tarafından oluşturulan kullanıcı iş yüklerinin yük dengelenmesi gibi fonksiyonları desteklediklerinden diğer SDDC işlevi için kritik bileşenler kadar kaynak tüketmezler. | Kaynak sıkıntısı durumunda NSX kenar cihazları SDDC kenar cihazlarından daha az kaynağa erişecekleri için takip ve kapasite yönetimi işleri ön etkin olarak gerçekleştirilmelidir. |
| SDDC-SA-VC-023 | Tüm kullanıcı sanal makineleri için oluşturulacak kaynak havuzu normal işlemci ve hafıza hisseleri ile yapılandırılacaktır. | Kaynak havuzu dışından oluşturulacak sanal makineler kaynak sıkıntısı durumunda diğer sanal makinelere sorun yaratacaktır. Paylaşımlı kenar ve işlem kümesinde SDDC kenar cihazları ağ erişimlerinin aksamaması için diğer tüm iş yüklerinden öncelikli şekilde kaynak garantisine sahip olmalıdır. | Kaynak sıkıntısı durumunda kullanıcı sanal makineleri kaynak dar boğazı yaşayabilirler ve kötü performans sergileyebilirler. Gerekirse sıkıntı yaşanmadan kenar cihazları için yeni kaynak eklenmeli ya da o iş için ayrılmış küme oluşturulmalıdır. |

Aşağıdaki tabloda paylaşımlı kenar ve işlem kümesinin mantıksal tasarımına ait özellik ve ayarlar özetlenmiştir. Küme içerisindeki sanal makine adedi az başlayıp, kullanıcı iş yükleri eklendikçe artmaya başlayacaktır:

Tablo 37 - Paylaşımlı Kenar ve İşlem Kümesi Mantıksal Tasarım Değerleri

|  |  |
| --- | --- |
| Özellik | Değer / Ayar |
| Paylaşımlı kenar ve işlem kümesini aşırı yüklenmeden destekleyebilecek sanallaştırma sunucu sayısı | 4 |
| Küme başına kaldırılabilen sanallaştırma sunucu arızası sayısı | 1 |
| Küme içerisinde kullanılabilen sanallaştırma sunucu sayısı | 3 |

* + - 1. İşlem Kümesi Tasarımı

SDDC büyüdükçe ek işlem kümeleri yapılandırılabilir. Misafir iş yükleri işlem kümelerinin üyesi olan ESXi sanallaştırma sunucuları üzerinde çalışırlar. Birden çok işlem kümesi işlem vCenter Server sistemi tarafından yönetilirler. Tasarım sanallaştırma sunucusu, kabin ilişkisini tarif eder ve vSphere HA ayarlarını belirler.

Tablo 38 - İşlem Kümesi Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-VC-024 | vSphere HA n+1 yedeklilik sağlayabilmek adına yüzde cinsinden yük devri kapasitesi yapılandırılacaktır. | Net sanallaştırma sunucusu yük devri kullanmak küme içerisindeki kullanılabilir kaynak miktarını sınırlar. | Küme içerisindeki bir sanallaştırma sunucusunun kaynakları rezerve edilmiştir. Bu da kaynaklar tükendiğinde yeni sanal makine oluşturulmasına engel olur. |

* + - 1. vCenter Server Özelleştirmeleri

vCenter Server gözlemleme, sanal makine arıza toleransı gibi pek çok geniş özelleştirme seçenekleri içerir. Her özellik için bu tasarımımızda ilgili tasarım kararları verilmiştir.

* + - 1. Sanal Makine ve Uygulama İzleme Hizmeti

Sanal makine ve uygulama izleme hizmeti etkinleştirildiğinde VMware tools yazılımı sayesinde küme içindeki her sanal makinenin çalıştığını denetler. Servis sanal makinelerin üzerinde çalışan VMware tools’dan gelen düzenli heartbeat’lerin varlığını ve I/O aktivitesi olup olmadığını kontrol eder. Eğer servis hiç bir heartbeat almaz ya da I/O aktivitesi tespit edemezse muhtemelen o sanal makinenin arızalandığına karar vererek yeniden başlatır.

Arızalanan, kilitlenen, çökmüş sanal makinelerin otomatik olarak yeniden başlatılabilmesi için sanal makine izleme servisi etkinleştirilmelidir. Sanal makinelerin barındırdıkları uygulamaya da servislerinde otomatik olarak başlayabilecek şekilde ayarlanmaları gerekmektedir.

Tablo 39 - Sanal Makine İzleme Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-VC-025 | Her küme için sanal makine izleme etkinleştirilmelidir. | Sanal makine izleme pek çok sanal iş yükü için yeterli bir koruma sağlamaktadır. | Sanal makine izleme servisinin etkinleştirilmesinin belirli bir dezavantajı bulunmamaktadır. |
| SDDC-SA-VC-026 | Yönetim ve paylaşımlı kenar ve işlem kümelerinde yeniden başlama kuralları için sanal makine grupları oluşturulacaktır. | Sanal makine grupları oluşturarak SDDC yönetim bileşenlerinin başlama sıralarını tarif eden kurallar yapılandırılabilir. | Bu grupların oluşturulması elle yapılan bir görev olup belli bir iş yükü getirmektedir. |
| SDDC-SA-VC-027 | SDDC yönetim bileşenlerinin yeniden başlama kuralları için sanal makine grupları oluşturulacaktır. | Bu kurallar sanal makinelerin doğru sırada başlamalarını sağlayarak SDDC yönetim bileşenlerinin sağlıklı olarak çalışmasını garanti eder. | Bu kuralların oluşturulması elle yapılan bir görev olup belli bir iş yükü getirmektedir. |

* + - 1. vSphere Dağıtık Kaynak Planlama (DRS)

vSphere DRS küme içerisindeki daha çok yüklü sanallaştırma sunucularından daha az olanlara iş yüklerini göç ettirerek kümenin yük dengelemesi işini gerçekleştirir. DRS elle ya da otomatik olarak yapılandırılabilir. Elle yapılandırma seçildiğinde DRS tavsiyelerde bulunur fakat sistem yöneticisi bu tavsiyelerin uygulanması için onay vermelidir. Otomatik seçenek yapılandırıldığında 5 ayrı seviyede olabilir. En düşük seviyede yeni sanal makinelerin yerleştirilmesi otomatik olarak yapılır ve sadece belirli kriterlerin yerine getirilmesi amacıyla (bakım kipine girilmeye çalışılması gibi) sanal makine iş yükleri göç ettirilir. En yüksek seviyede ise en ufak bir yük dengelenmesinde fayda görüleceği hesaplandığında sanal makine iş yükleri göç ettirilebilir.

Tablo 40 - vSphere DRS Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-VC-028 | Tüm kümelerde DRS etkinleştirilecek, var sayılan değeri olan “medium” ile tam otomatik olarak yapılandırılacaktır. | Var sayılan ayarlar yük dengeleme ile aşırı vMotion olayı gerçekleşmesi arasında ideal bir denge sağlar. | Bir vCenter server kesintisi esnasında sanal makinelerin hangi ESXi sanallaştırma sunucusunda olduğunun belirlenmesi zor olabilir. |

* + - 1. Gelişmiş vMotion Uyumluluğu (EVC)

EVC daha yeni işlemcilerin bazı özelliklerini maskeleyerek eski işlemciye sahip sanallaştırma sunucularına göç yapılabilmesini sağlar. EVC sadece aynı üreticinin işlemcileri arasında çalışmaktadır ve işlemci ailelerinin arasındaki sürüm farklarının de bir sınırı bulunmaktadır.

Eğer küme oluşturulurken EVC ayarlanırsa sonradan daha yeni işlemciye sahip sanallaştırma sunucularının kesintiye gerek duymadan eklenmesi mümkün olur. EVC’nin mevcut işlemci için desteklenen en yüksek seviyesi ayarlanmalıdır.

Tablo 41 - EVC Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-VC-029 | Tüm kümelerde EVC etkileştirilmelidir. Küme içerisindeki tüm sanallaştırma sunucularının desteklediği en düşük seviye ayarlanmalıdır. | Sanal makinelerde kesinti yaşamadan tüm kümenin güncellenmesini mümkün kılar. | EVC’nin etkinleştirilebilmesi sadece küme içindeki sanallaştırma sunucuların işlemcileri hep aynı üreticiden sağlanmışsa mümkün olur. |

* + - 1. Transport Layer Security (TLS) Sertifikaları Kullanımı

Varsayılan kurulum ile vSphere 6.5 VMCA (VMware Certificate Authority) tarafından imzalanmış TLS/SSL sertifikaları kullanır. Bu sertifikalar son kullanıcı cihazları ya da internet tarayıcıları tarafından güvenilmezler. En azından tüm kullanıcılara hitap eden sertifikaların üçüncü parti ya da kurumsal sertifika otoritesi tarafından imzalanmış sertifikalar ile değiştirilmesi güvenlik açısından en iyi çalışma prensiplerindendir. Sunucudan sunucuya olan iletişim için VMCA imzalanmış sertifikalar kullanılmaya devam edebilir.

Tablo 42 - vCenter Server TLS Sertifika Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-VC-030 | vCenter Server ve platform hizmetleri kontrolcüsü sanal makineleri sertifikaları üçüncü parti PKI tarafından imzalanmış sertifikalar ile değiştirilecektir. | Alt yapı yöneticileri vCenter Server’a ve platform hizmetleri kontrolcüsüne tarayıcı vasıtasıyla bağlanarak yapılandırma ve izleme yapmaktadırlar. Varsayılan sertifikalar ile uyarı hataları alınmaktadır. | Sertifikaların değiştirilmesi ve yönetimi fazladan operasyonel iş yükü getirmektedir. |
| SDDC-SA-VC-031 | Sertifikalar imzalanırken SHA-2 ya da daha güvenli bir algoritma kullanılmalıdır. | SHA-1 algoritması güvensiz sayıldığından güncelliğini yitirmiştir. | Tüm sertifika otoriteleri SHA-2 algoritmasını desteklemeyebilir. |

* + 1. Sanallaştırma Ağı Tasarımı

İyi tasarlanmış bir ağ, organizasyonun iş hedeflerine erişmesine yardımcı olur. Yetkisiz erişimleri engeller ve iş ile ilgili verilere zamanında erişilmesini sağlar.

Bu ağ sanallaştırma tasarımı sanal ağ için vSphere ve VMware NSX yazılımlarını kullanmaktadır. Bu tasarım içerisinde üst seviye ağ tasarım yöntemlerinden ve en iyi uygulama yöntemleri anlatılmaktadır. Ayrıca ağ tasarımı içerisinde ana başlıklar olarak sanal anahtarlar, NIC birleştirme, ağ I/O kontrolü, VXLAN ve vMotion TCP/IP yığını hakkında da bilgi ve tasarım kararları verilecektir.

* + - 1. Sanal Ağ Tasarım Yönlendirici İlkeleri

Bu tasarım üst seviye ağ tasarım ilkelerini takip eder ve ağ ile ilgili en iyi uygulama yöntemlerini benimser. Üst seviye tasarım hedefleri ortamdan bağımsız olarak uygulanır. Tasarım hedefleri aşağıdaki gibidir:

* Muhtelif ihtiyaçlara karşılık verebilmelidir. Ağ organizasyon içerisindeki birbirinden farklı varlıkların çeşitli ihtiyaçlarını giderebilmelidir. Bu varlıklar uygulamaları, hizmetleri, veri depolamayı, yöneticileri ve kullanıcıları içerir.
* Maliyetlerin azaltılması. vSphere altyapısında erişilmek istenen hedeflerden birisi de maliyetlerin azaltılmasıdır. Sadece sunucuların konsolidasyonu bile gerekli bağlantı sayılarının ve karşılık verecek cihazların sayılarının azalacağı için maliyetleri düşürür. Bununlar birlikte daha efektif bir ağ yapısı tercih edilecektir. Örnek vermek gerekirse VLAN’lara ayrılmış iki adet 10GbE NIC bir düzine fiziksel olarak ayrı yapılandırılmış 1GbE bağlantıdan daha maliyet etkin olacaktır.
* Performansın artırılması. Gerekli miktarda bant genişliği sağlanması ki sıkışma ve gecikmeleri azaltırken, performansı artırabilir ve bakım yapma sıklığını azaltabilir.
* Erişilebilirliğin geliştirilmesi. İyi tasarlanmış bir ağ genellikle yedeklilik sağlamak suretiyle erişilebilirliği artırır.
* Güvenliğin desteklenmesi. İyi tasarlanmış bir ağ kontrollü erişim (gerektiği yerde) ve izolasyon (ihtiyaç olan yerde) aracılığıyla kabul edilebilir seviyede güvenlik sağlanmasını desteklemelidir.
* Altyapı fonksiyonlarının geliştirilmesi. Ağın, vMotion, HA Fault Tolerance gibi vSphere özelliklerini destekleyecek şekilde yapılandırılması gereklidir.

Ağ için en iyi uygulama yöntemleri aşağıdaki gibidir:

* Daha yüksek seviyede güvenlik sağlayabilmek ve daha yüksek performans için ağ servislerinin birbirlerinden ayrılması gereklidir.
* Kritik sanal makinelere bant genişliğinin garanti edilebilmesi için ağ I/O kontrolü ve trafik şekillendirme yapılmalıdır. Ağ kaynaklarının azaldığı durumlarda bu sanal makinelere daha fazla oranda bant genişliği tahsis edilecektir.
* Farklı VLAN ID’lerine sahip port gruplarına bağlayarak tek bir vSphere Distributed Switch üzerinde ağ hizmetleri ayrılmalıdır.
* vSphere vMotion trafiğini ayrı bir ağ üzerinde tutmak gerekir. vMotion ile göç esnasında sanal makinenin hafızası ağ üzerinden transfer edilmektedir. Adanmış bir vMotion VLAN’ı kullanarak ayrı bir ağ kullanılmalıdır.
* Linux kernel sürümü 2.6.20 ya da daha önceki bir işletim sistemi ile düz geçiş cihazlar kullanılırken MSI ve MSI-X kiplerinden kaçınmak gerekmektedir. Bu kiplerin kayda değer negatif performans etkileri bulunmaktadır.
* Daha yüksek performans alabilmek için tüm sanal makinelerin sanal NIC sürücüleri için mümkün olduğunca VMXNET3 kullanılmalıdır.
* Aynı vSphere standart switch ya da distributed switch’e bağlı olan fiziksel ağ adaptörlerinin aynı fiziksel ağa da bağlı olduğundan emin olunmalıdır.

Farklı tipteki ağ trafiklerinin ayrılması çekişmeyi ve yüksek gecikme değerlerini engellemek için gereklidir. Erişim güvenliği sağlamak için de ağların ayrılması gerekmektedir.

Herhangi bir ağdaki yüksek gecikme değerleri performansı negatif yönde etkiler. Örnek vermek gerekirse gecikmenin azaltılması IP tabanlı veri depolamada ve vSphere fault tolerance loglama ağındaki önemlidir çünkü bu ağlarda gerçekleşebilecek gecikmeler birden çok sanal makinenin performansını kötü yönde etkileyecektir.

Uygulama ya da hizmete bağlı olarak belirli sanal makineler üzerindeki yüksek gecikme değerleri performansı negatif yönde etkiler. Mevcut duruma dair durum analiz bilgilerinin toplanarak ve hizmet sahiplerinin de katkıları ile gecikmeye karşı hassas olan iş yüklerinin belirlenmesi gerekmektedir.

Trafik tipine bağlı olarak gereken ağ ve VLAN’ların sayısına karar vermek gereklidir. Bu trafik tipleri ana olarak vSphere operasyonel trafiği (yönetim, vMotion, vSAN, NFS veri depolama, vSphere replikasyonu ve VXLAN) ve organizasyonun servis ve uygulamalarının ihtiyaç duyduğu ağ trafiğidir.

* + - 1. Sanal Anahtarlar

Sanal anahtarlar, sanal ağ yönetimi işlemleri gerçekleştirirken tek bir noktadan izleme ve yapılandırma sağlayarak işleri kolaylaştırırlar. Bir vSphere distributed switch standart bir sanal switch’e kıyasla daha gelişkin özellikler sağlar. Merkezi olarak vCenter Server tarafından oluşturulup yönetildiklerinden dolayı anahtar yapılandırmaları tüm ESXi sanallaştırma sunucuları arasından tutarlı bir şekilde gerçekleştirilebilir. Merkezi yönetim zamandan kazandırır, hata yapılmasını engeller ve operasyonel maliyetlerin azalmasını sağlar. Distributed switch’lerde standart sanal anahtarlarda bulunmayan bazı özellikler bulunur. Bu özelliklerin bazıları organizasyonun altyapısında çalışan servisler ve uygulamalar için faydalı olabilir. Örnek vermek gerekirse NetFlow ve port aynalama gibi özellikler sanal altyapının izlenmesinde ve sorun giderilmesinde fayda sağlar. Bütün bunlarla birlikte vCenter Server hizmet dışı kalırsa distributed switchler yönetilemezler. Bu sebeple vCenter Server 1. Seviye bir uygulama sayılmalıdır.

Sağlık kontrolü hizmeti bir vSphere distributed switch üzerinde olabilecek sorunların tespit edilip giderilmesi için yardımcı olur. Aşağıda sağlık kontrolü hizmetinin tespit edebileceği sıradan yapılandırma hataları verilmiştir:

* ESXi sanallaştırma sunucuları ile fiziksel anahtarlar arasında birbirlerine uymayan VLAN trunkları.
* Fiziksel ağ adaptörleri, distributed switch’ler ve fiziksel anahtar portları arasındaki birbirlerine uymayan MTU ayarları.
* Fiziksel anahtar port channel ayarları ile bunlara uymayan sanal switch politikaları.

Sağlık kontrolü ESXi sanallaştırma sunucunun ağ adaptörünün bağlı olduğu port ile sınırlıdır.

Mümkün olduğunca az sayıda sanal anahtar kullanılmalı, tercihen sadece bir adet sanal anahtar olmalıdır. Her tip ağ trafiği için, yapılandırma ve izlemenin kolaylaştırılması adına tek bir port grup yapılandırılmalıdır.

Tablo 43 - Sanal Anahtar Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-AĞ-001 | Tüm sanal distributed switchler üzerinde sağlık kontrolü açılmalıdır. | Sağlık kontrolü tüm sanallaştırma sunucularına bağlı distributed switchlere tüm VLAN’ların trunklandığını ve fiziksel ağ ile aynı MTU değerinin ayarlandığını kontrol eder. | Bu özelliğin kullanılabilmesi için en az iki adet fiziksel bağlantı olması gerekir. |
| SDDC-SA-AĞ-002 | vSphere Distributed Switch (VDS) kullanılmalıdır. | VDS kullanımı yönetimi kolaylaştırır. | Yedekliliğin sağlanabilmesi için VSS’den VDS’e göç için en az iki adet fiziksel bağlantı gerekir |
| SDDC-SA-AĞ-003 | Her bir küme için tek bir VDS kullanılmalıdır. | Ağ tasarımını basitleştirir. Hata etki alanını azaltır. | Yönetilmesi gereken VDS sayısını arttırır. |

* + - 1. Yönetim Kümesi Distributed Switchleri

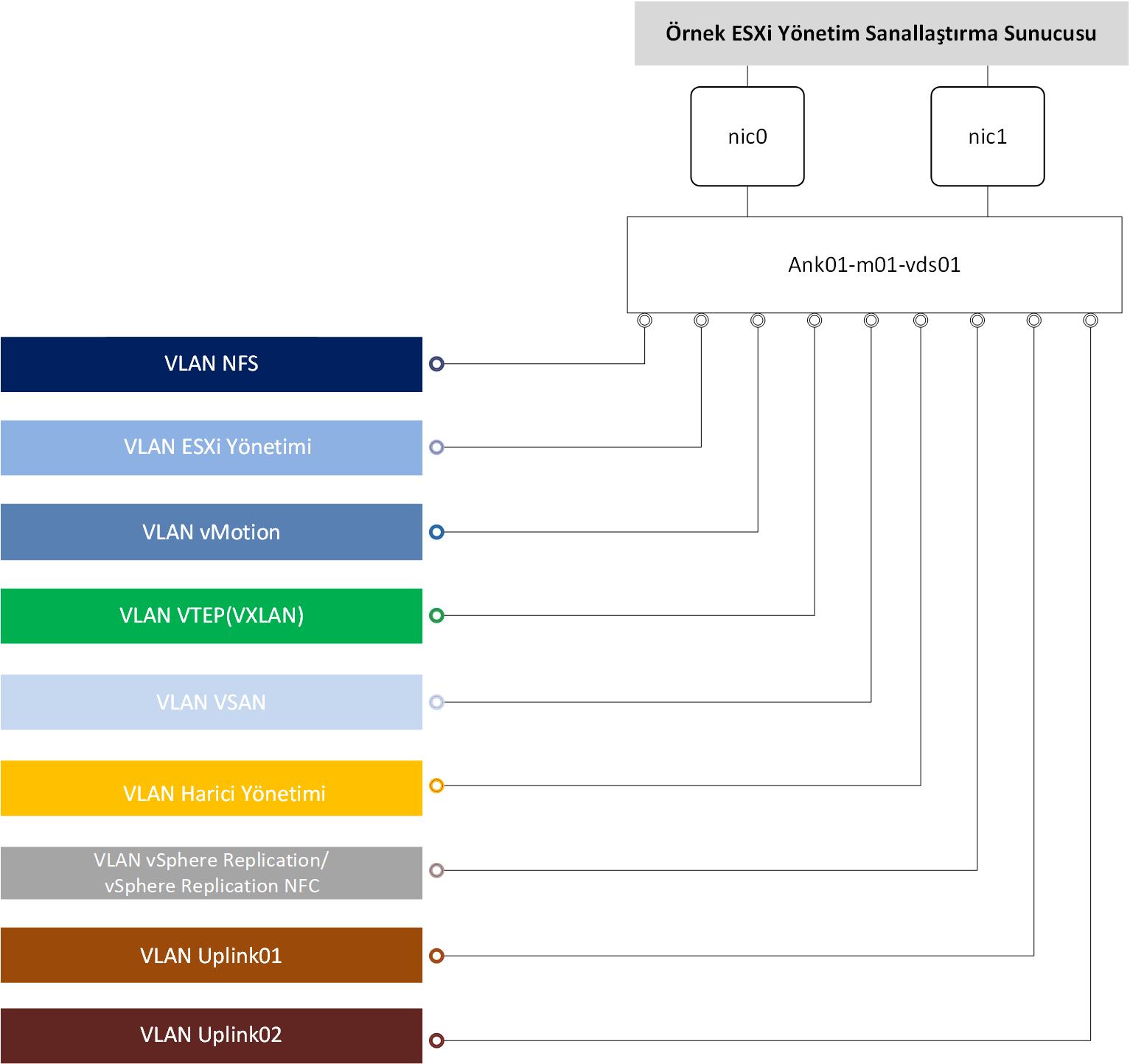
Yönetim kümesi aşağıdaki ayarlara sahip tek bir VDS kullanır:

Tablo 44 - Yönetim Kümesi için Sanal Anahtar

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| VDS Adı | Fonksiyonu | Ağ I/O Kontrolü | Fiziksel Port sayısı | MTU |
| Ank01-m01-vds01 | * ESXi Yönetimi * NFS * vSAN * vSphere vMotion * VXLAN Tunnel Endpoint (VTEP) * vSphere Replication * ECMP’yi etkinleştirebilmek için 2 adet uplink * Harici yönetim bağlantıları | Etkin | 2 | 9000 |

Tablo 45 - vDS-MgmtPort Grubu Yapılandırma Ayarları

|  |  |
| --- | --- |
| Parametre | Değer /Ayar |
| Failover Detection | Link Status only |
| Notify Switches | Enabled |
| Failback | Yes |
| Failover Order | Active Uplinks, Uplink1, Uplink2 |



Şekil 3 - Yönetim Sanallaştırma Sunucusu için Ağ Anahtarı Tasarımı

Tasarımın bu kısmında fiziksel NIC yerleşimi ve fiziksel ağ özellikleri gösterilerek mantıksal ağ tasarımı genişletilmektedir.

Tablo 46 - Yönetim Sanal Anahtarları, Fiziksel/Sanal NIC

|  |  |  |
| --- | --- | --- |
| VDS | vmnic | Fonksiyon |
| Ank01-m01-vds01 | 0 | uplink |
| Ank01-m01-vds01 | 1 | uplink |

Tablo 47 - Yönetim Sanal Anahtarları, Port Grupları ve VLAN'ları

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| VDS | Port Grup Adı | Teaming Politikası | Aktif Uplinkler | VLAN ID |
| Ank01-m01-vds01 | Ank01-m01-vds01-management | Fiziksel NIC yüküne göre yönlendirilecek. | 0,1 | 1611 |
| Ank01-m01-vds01 | Ank01-m01-vds01-vmotion | Fiziksel NIC yüküne göre yönlendirilecek. | 0,1 | 1612 |
| Ank01-m01-vds01 | Ank01-m01-vds01-vsan | Fiziksel NIC yüküne göre yönlendirilecek. | 0,1 | 1613 |
| Ank01-m01-vds01 | Otomatik oluşacak (VTEP) | SRC-ID ye göre yönlendirilecek. | 0,1 | 1614 |
| Ank01-m01-vds01 | Ank01-m01-vds01-uplink01 | Fiziksel NIC yüküne göre yönlendirilecek. | 0,1 | 2711 |
| Ank01-m01-vds01 | Ank01-m01-vds01-uplink02 | Fiziksel NIC yüküne göre yönlendirilecek. | 0,1 | 2712 |
| Ank01-m01-vds01 | Ank01-m01-vds01-nfs | Fiziksel NIC yüküne göre yönlendirilecek. | 0,1 | 1615 |
| Ank01-m01-vds01 | Ank01-m01-vds01-replication | Fiziksel NIC yüküne göre yönlendirilecek. | 0,1 | 1616 |
| Ank01-m01-vds01 | Ank01-m01-vds01-ext-management | Fiziksel NIC yüküne göre yönlendirilecek. | 0,1 | 130 |

Tablo 48 - Yönetim VMkernel Adaptörleri

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| VDS | Ağ Adı | Bağlı Port Grubu | Verilen Servisler | MTU |
| Ank01-m01-vds01 | Management | Ank01-m01-vds01-management | Yönetim trafiği | 1500 (varsayılan) |
| Ank01-m01-vds01 | vMotion | Ank01-m01-vds01-vmotion | vMotion trafiği | 9000 |
| Ank01-m01-vds01 | vSAN | Ank01-m01-vds01-vsan | vSAN trafiği | 9000 |
| Ank01-m01-vds01 | NFS | Ank01-m01-vds01-nfs | - | 9000 |
| Ank01-m01-vds01 | Replication | Ank01-m01-vds01-replication | vSphere replication trafiği | 9000 |
| Ank01-m01-vds01 | VTEP | Otomatik oluşacak (NSX VTEP) | - | 9000 |

* + - 1. Paylaşımlı Kenar ve İşlem Kümesi Distributed Switchleri

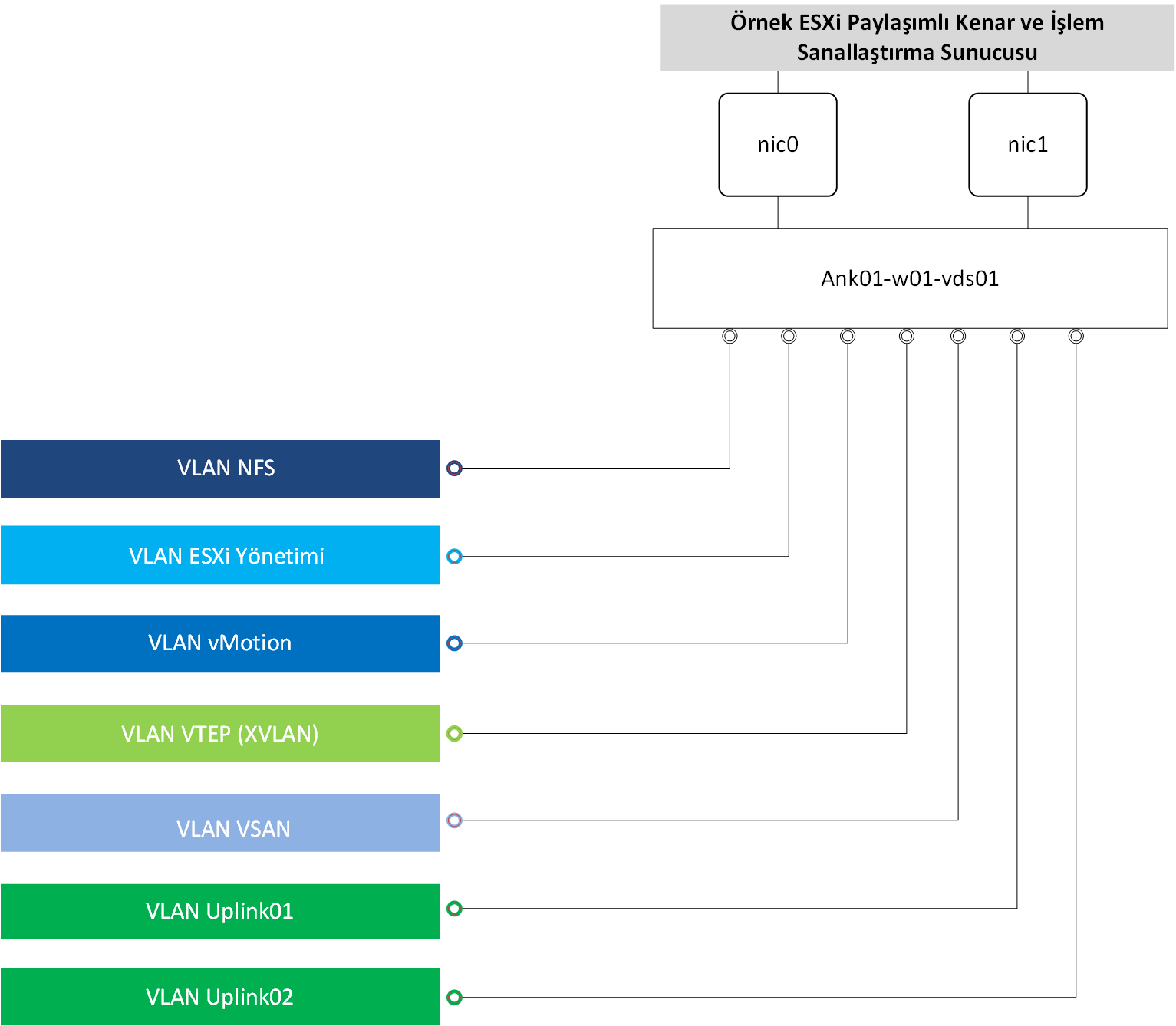
Paylaşımlı kenar ve işlem kümesi aşağıdaki ayarlara sahip tek bir VDS kullanır:

Tablo 49 - Paylaşımlı Kenar ve İşlem Kümesi Sanal Anahtarı

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| VDS Adı | Fonksiyonu | Ağ I/O Kontrolü | Fiziksel Port sayısı | MTU |
| Ank01-w01-vds01 | * ESXi Yönetimi * NFS * vSphere vMotion * VXLAN Tunnel Endpoint (VTEP) * ECMP’yi etkinleştirebilmek için 2 adet uplink * vSAN * Harici müşteri/kiracı bağlantısı | Etkin | 2 | 9000 |

Tablo 50 - vDS-Comp01 Port Grubu Yapılandırma Ayarları

|  |  |
| --- | --- |
| Parametre | Değer /Ayar |
| Failover Detection | Link Status only |
| Notify Switches | Enabled |
| Failback | Yes |
| Failover Order | Active Uplinks, Uplink1, Uplink2 |



Şekil 4 - Paylaşımlı Kenar ve İşlem Küme Ağ Anahtar Tasarımı

Tablo 51 - Paylaşımlı Kenar ve İşlem Kümesi Sanal Anahtarları, Fiziksel/Sanal NIC

|  |  |  |
| --- | --- | --- |
| VDS | vmnic | Fonksiyon |
| Ank01-w01-vds01 | 0 | uplink |
| Ank01-w01-vds01 | 1 | uplink |

Tablo 52 - Paylaşımlı Kenar ve İşlem Kümesi Sanal Anahtarları, Port Grupları ve VLAN'ları

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| VDS | Port Grup Adı | Teaming Politikası | Aktif Uplinkler | VLAN ID |
| Ank01-w01-vds01 | Ank01-w01-vds01-management | Fiziksel NIC yüküne göre yönlendirilecek. | 0,1 | 1631 |
| Ank01-w01-vds01 | Ank01-w01-vds01-vmotion | Fiziksel NIC yüküne göre yönlendirilecek. | 0,1 | 1632 |
| Ank01-w01-vds01 | Ank01-w01-vds01-vsan | Fiziksel NIC yüküne göre yönlendirilecek. | 0,1 | 1633 |
| Ank01-w01-vds01 | Ank01-w01-vds01-nfs | Fiziksel NIC yüküne göre yönlendirilecek. | 0,1 | 1615 |
| Ank01-w01-vds01 | Otomatik oluşacak (NSX VTEP) | SRC-ID ye göre yönlendirilecek | 0,1 | 1634 |
| Ank01-w01-vds01 | Ank01-w01-vds01-uplink01 | Fiziksel NIC yüküne göre yönlendirilecek. | 0,1 | 1635 |
| Ank01-w01-vds01 | Ank01-w01-vds01-uplink02 | Fiziksel NIC yüküne göre yönlendirilecek. | 0,1 | 2713 |

Tablo 53 - Paylaşımlı Kenar ve İşlem Kümesi VMkernel Adaptörleri

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| VDS | Ağ Adı | Bağlı Port Grubu | Verilen Servisler | MTU |
| Ank01-w01-vds01 | Management | Ank01-w01-vds01-management | Yönetim trafiği | 1500 (varsayılan) |
| Ank01-w01-vds01 | vMotion | Ank01-w01-vds01-vmotion | vMotion trafiği | 9000 |
| Ank01-w01-vds01 | vSAN | Ank01-w01-vds01-vsan | vSAN trafiği | 9000 |
| Ank01-w01-vds01 | NFS | Ank01-w01-vds01-nfs | - | 9000 |
| Ank01-w01-vds01 | VTEP | Otomatik oluşacak (NSX VTEP) | - | 9000 |

* + - 1. İşlem Kümesi Dağıtık Anahtarları

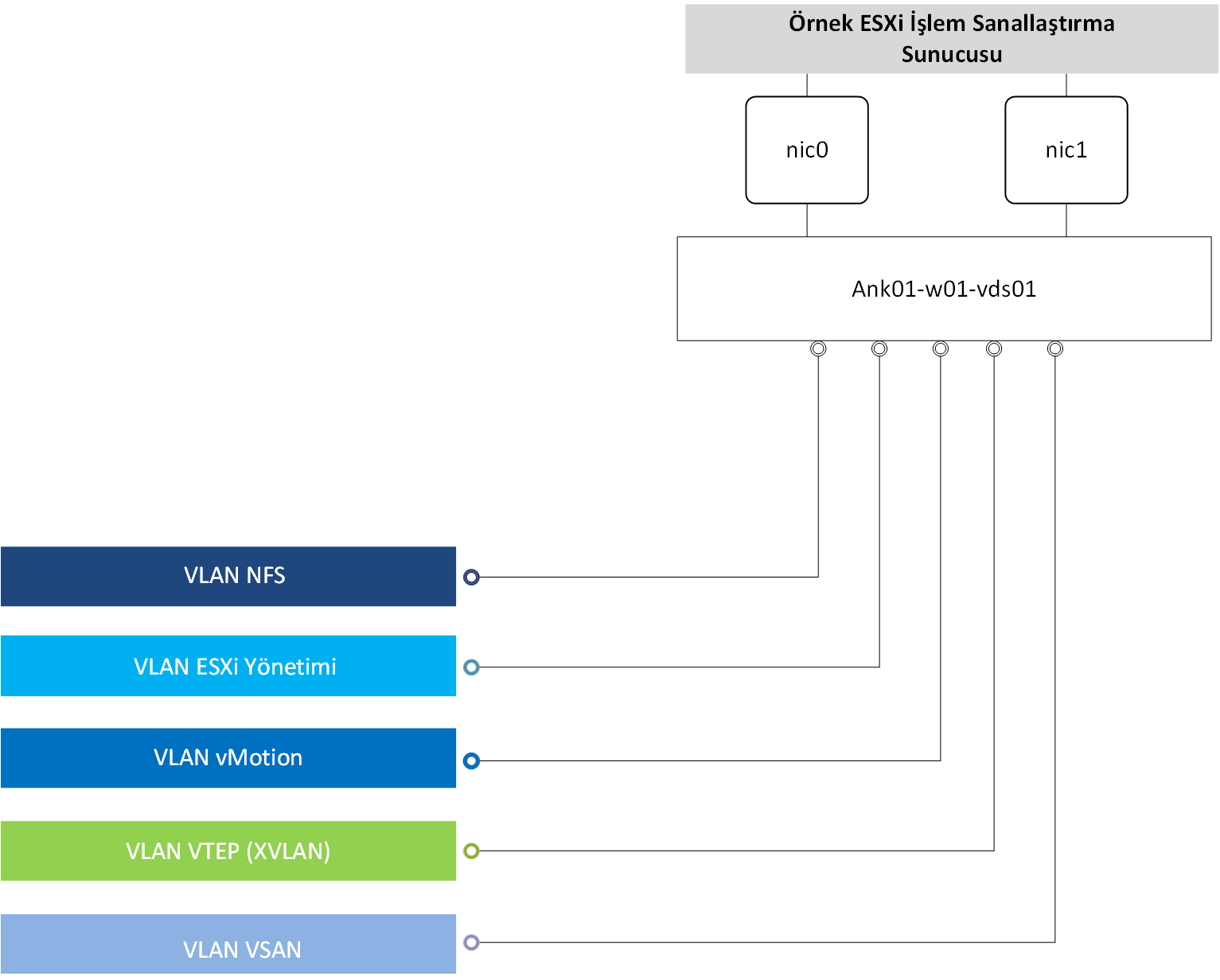
İşlem kümesi için olan vSphere Distributed Switch aşağıdaki ayarlar ile yapılandırılır:

Tablo 54 - Adanmış İşlem Kümesi için Sanal Anahtarı

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| VDS | Fonksiyon | Ağ I/O Kontrolü | Fiziksel Port Sayısı | MTU |
| Ank01-w01-vds02 | * ESXi Yönetimi * NFS * vSphere vMotion * VXLAN Tunnel Endpoint (VTEP) | Etkin | 2 | 9000 |

Tablo 55 - vDS-Comp02 Port Grubu Yapılandırma Ayarları

|  |  |
| --- | --- |
| Parametre | Değer /Ayar |
| Failover Detection | Link Status only |
| Notify Switches | Enabled |
| Failback | Yes |
| Failover Order | Active Uplinks, Uplink1, Uplink2 |



Şekil 5 - İşlem Kümeleri için Ağ Anahtarı Tasarımı

Tablo 56 - İşlem Kümesi Sanal Anahtarları, Fiziksel/Sanal NIC

|  |  |  |
| --- | --- | --- |
| VDS | vmnic | Fonksiyon |
| Ank01-w01-vds02 | 0 | Uplink |
| Ank01-w01-vds02 | 1 | Uplink |

Tablo 57 - İşlem Kümesi Sanal Anahtar Port Grupları ve VLAN'ları

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| VDS | Port Grup Adı | Teaming Politikası | Aktif Uplinkler | VLAN ID |
| Ank01-w01-vds02 | Ank01-w01-vds02-management | Fiziksel NIC yüküne göre yönlendirilecek. | 0,1 | 1621 |
| Ank01-w01-vds02 | Ank01-w01-vds02-vmotion | Fiziksel NIC yüküne göre yönlendirilecek. | 0,1 | 1622 |
| Ank01-w01-vds02 | Otomatik oluşacak (NSX VTEP) | SRC-ID ye göre yönlendirilecek | 0,1 | 1624 |
| Ank01-w01-vds02 | Ank01-w01-vds02-nfs | Fiziksel NIC yüküne göre yönlendirilecek. | 0,1 | 1625 |

Tablo 58 - İşlem Kümesi VMkernel adaptörleri

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| VDS | Ağ Adı | Bağlı Port Grubu | Verilen Servisler | MTU |
| Ank01-w01-vds02 | Management | Ank01-w01-vds02-management | Yönetim trafiği | 1500 (varsayılan) |
| Ank01-w01-vds02 | vMotion | Ank01-w01-vds02-vmotion | vMotion trafiği | 9000 |
| Ank01-w01-vds02 | NFS | Ank01-w01-vds02-nfs | - | 9000 |
| Ank01-w01-vds02 | VTEP | Otomatik oluşacak (NSX VTEP) | - | 9000 |

* + - 1. NIC Birleştirme (Teaming)

Bir ağ yolu üzerindeki bant genişliğinin artırılabilmesi ve daha yüksek erişilebilirlik sağlayan yedekliliğin sağlanabilmesi için NIC birleştirme yapılabilir.

NIC birleştirme tek bir noktadan kaynaklanabilecek arızaların gerçekleşmesini önler ve trafik yükün dağıtılabilmesi için seçenekler sunar. Tek noktadan kaynaklanabilecek arızaların daha da önüne geçilebilmesi için birden fazla kart ve ana kartta bulunan NIC’ler (kart ve port yedekliliği) ile birleştirme yapılabilir.

Ayrı fiziksel anahtarlara bağlanan birleştirilmiş NIC’ler ile tek bir sanal anahtar kullanılmalıdır.

Bu tasarımda fiziksel NIC yüküne göre yönlendirme yapan aktif-aktif yapılandırma kullanılmaktadır. Bu yapılandırmada boş NIC’ler arıza gerçekleşip yük devralmayı beklemez, bant genişliğine katkıda bulunurlar.

Tahmin edilebilir seviyede performans sağlayabilmek için aşağıda gösterilen yapılandırma ile birden çok ağ kartı kullanılmalıdır:

* İki ayrı anahtara bağlanılan ve açıkça yük devralma sırasının belirlendiği aktif pasif yapılandırma.
* İki ya da daha fazla fiziksel ağ kartının aktif olarak rol aldıkları aktif aktif yapılandırma.

Bu tasarım içerisinde aktif aktif yapılandırma kullanılmaktadır.

Tablo 59 NIC Birleştirme ve Politikaları

|  |  |  |  |
| --- | --- | --- | --- |
| Tasarım Kalitesi | Aktif Aktif | Aktif Pasif | Yorumlar |
| Erişilebilirlik | Olumlu | Olumlu | Birleştirme yapılması seçeneklerden bağımsız olarak ortamın erişilebilirliğini artırır. |
| Yönetilebilirlik | Etkisi yok | Etkisi yok | İki tasarımın da yönetilebilirliğe etkisi bulunmamaktadır. |
| Performans | Olumlu | Etkisi yok | Aktif aktif tasarımda her iki ağ kartından da trafik gönderilmektedir. Dolayısıyla bant genişliği artmaktadır. Bu yapılandırmada eğer ağ kartları farklı trafik tipleri arasında kullanılıyorsa ve ağ I/O kontrolü yapılıyorsa faydalıdır. |
| Kurtarılabilirlik | Etkisi yok | Etkisi yok | İki tasarımın da yönetilebilirliğe etkisi bulunmamaktadır. |
| Güvenlik | Etkisi yok | Etkisi yok | İki tasarımın da yönetilebilirliğe etkisi bulunmamaktadır. |

Tablo 60 - NIC Birleştirme Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-AĞ-004 | VXLAN trafiği taşıyan port grupları haricindeki tüm port grupları için Fiziksel NIC yüküne göre yönlendirme algoritması kullanılacaktır. VTEP kernel portları ve VXLAN trafiği SRC-ID ye göre yönlendirme kuralını kullanacaktır. | Ağ tasarımını basitleştirirken direnç ve performans artar. | NSX fiziksel NIC yüküne göre yönlendirmeyi desteklemediği için iki ayrı algoritma kullanımı zorunludur. |

* + - 1. Ağ I/O Kontrolü

Ağ I/O kontrolü etkinleştirildiği zaman, distributed switch aşağıdaki sistem trafik tiplerine bant genişlikleri tahsis eder:

* Fault Tolerance
* iSCSI Trafiği
* vSphere vMotion trafiği
* Yönetim trafiği
* vSphere Replication trafiği
* NFS trafiği
* vSAN trafiği
* vSphere Data Protection yedekleme trafiği
* Sanal makine trafiği

Ağ I/O kontrolü sadece bant genişliğinde bir sıkışma olduğunda farklı trafik tiplerine ayrılmış hisse oranlarını zorunlu olarak tatbik eder. Bant genişliğinde bir sıkışma olduğunda her trafik tipine hisse oranları tahsis eder. Bunun sonucunda hisse oranından belirlenen daha az önemli trafik yavaşlatılır ve daha önemli trafik tiplerinin daha fazla ağ kaynaklarına erişmesi sağlanır.

Ağ I/O kontrolü aynı zamanda bir sanallaştırma sunucusu üzerindeki fiziksel adaptörlerin kapasitelerine dayanarak sistem trafiği için rezervasyonlar sağlayarak sanal makine ağı adaptörü seviyesinde ince ayar yapılabilmesini sağlar. Kaynak kontrolü vCenter’da bulunan CPU ve hafıza rezervasyonları ile benzerdir.

Tablo 61 - Ağ I/O Kontrolü Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-AĞ-005 | Tüm dsitributed switchler üzerinde ağ I/O kontrolü açılacaktır. | Ağın direnci ve performansı artar. | Eğer yanlış yapılandırılırsa ağ I/O kontrolü kritik trafik tipleri için performansı kötü yönde etkileyebilir. |
| SDDC-SA-AĞ-006 | vMotion trafiği için hisse değeri düşük olarak ayarlanacaktır. | Sıkışıklık anlarında vMotion trafiği sanal makine trafiği ya da veri depolama trafiği kadar önemli değildir. | Sıkışıklık esnasında vMotion süreleri normalden fazla uzayacaktır. |
| SDDC-SA-AĞ-007 | vSphere Replication trafiği için hisse değeri düşük olarak ayarlanacaktır. | Sıkışıklık anlarında replikasyon trafiği sanal makine trafiği ya da veri depolama trafiği kadar önemli değildir. | Sıkışıklık esnasında replikasyon normalden daha uzun sürecektir ve tanımlı SLA’leri ihlal edebilir. |
| SDDC-SA-AĞ-008 | vSAN trafiği için hisse değeri yüksek olarak ayarlanacaktır. | Sıkışıklık anlarından sanal makine performansının etkilenmemesi için vSAN trafiği için gerekli olan bant genişliğinin garantilenmesi gerekir. | Yoktur. |
| SDDC-SA-AĞ-009 | Yönetim ağı trafiği için hisse değeri normal olarak ayarlanacaktır. | Varsayılan ayarların korunarak bırakılması yönetim trafiğinin vMotion ve replikasyondan yüksek, vSAN’dan düşük olarak önceliklendirilmesini sağlar. Sıkışıklık durumunda dahi sistemin yönetilmesini sağlar. | Yoktur. |
| SDDC-SA-AĞ-010 | NFS trafiği için hisse değeri düşük olarak ayarlanacaktır. | VDP yedekleri ve Log Insight arşivleri için kullanılacağından, NFS ikincil öneme saihp bir veri depolama sistemidir. Bu nedenler vSAN’dan daha düşük bir öncelik verilmiştir. | Sıkışıklık zamanlarından VDP yedekleri normalden daha uzun sürede bitebilir. |
| SDDC-SA-AĞ-011 | vSphere Data Protection yedekleme trafiği için hisse değeri düşük olarak ayarlanacaktır. | Sıkışıklık esnasında ana fonksiyonların ağ kaynaklarıne erişmesi daha önemli olduğu için yedekleme trafiğine göre daha yüksek önceliğe sahiptirler. | Sıkışıklık zamanlarından VDP yedekleri normalden daha uzun sürede bitebilir. |
| SDDC-SA-AĞ-012 | Sanal makinelerin hisse değerleri yüksek olarak ayarlanacaktır. | SDDC içerisindeki en önemli varlık sanal makinelerdir. Yüksek erişim hakkı tanıyarak her zaman sanal makinelerin ihtiyaçları olan ağ kaynaklarına erişebileceklerini garanti etmeye çalışırız. | Yoktur. |
| SDDC-SA-AĞ-013 | Fault Tolerance trafiği için hisse değeri düşük olarak ayarlanacaktır. | Bu tasarımda Fault tolerance kullanılmadığından en düşük öneme sahiptir. | Yoktur. |
| SDDC-SA-AĞ-014 | iSCSI trafiği için hisse değeri düşük olarak ayarlanacaktır. | Bu tasarımda iSCSI kullanılmadığından en düşük öneme sahiptirler. | Yoktur. |

* + - 1. VXLAN

VXLAN veri merkezleri dokuları arasında izole, birden çok sakin barındırabilen yayın alanları oluşturma kabiliyetini ve müşterilerin esnek, fiziksel ağ sınırlarına yayılan mantıksal ağlar oluşturulmasını sağlar.

Bu mantıksal ağların oluşturulmasındaki ilk adım, ağ kaynaklarının soyutlanarak bir havuzda birleştirilmesidir. Nasıl vSphere işlem kapasitelerini sunucu donanımından soyutlayıp kaynak havuzları haline getirerek hizmet olarak tüketilecek hale getiriyorsa, distributed switch ve VXLAN da ağı genel bir ağ kapasitesi havuzuna soyutlayarak ve bu servislerin tüketimini altında yatan fiziksel yapıdan ayırır. Bir ağ kapasitesi havuzu fiziksel sınırlara yayılabilir, kümeler ve coğrafik olarak ayrık veri merkezleri arasında kullanımı optimize edebilir. Tümleşik ağ kapasite havuzu ideal olarak belirli uygulamalara direk olarak bağlı mantıksal ağlara bölünebilir.

VXLAN standart Layer 3 IP paketleri içerisinde enkapsüle edilmiş Layer 2 mantıksal ağlar oluşturarak çalışır. VLAN etiketlerine ihtiyaç duymadan her frame içerisinde bulunan bölüm id’leri sayesinde VXLAN mantıksal ağları birbirlerinden farklılaşırlar. Sonuç olarak büyük miktarda izole Layer 3 VXLAN ağları sıradan Layer 3 alt yapısı üzerinde bir arada var olabilirler.

vSphere mimarisinde enkapsüle edilme işlemi sanal makinenin sanal NIC’i ile sanal anahtarın mantıksal portu arasında gerçekleşir. Buda VXLAN’ı hem sanal makineye hem de altında yatan Layer 3 ağa şeffaf kılar. VXLAN ve VXLAN olmayan sunucular (fiziksel bir başka sunucu ya da internet yönlendiricisi gibi) arası geçiş hizmeti NSX Edge Services Gateway tarafından gerçekleştirilir. Edge geçiti VXLAN bölüm id’lerini VLAN id’lerine çevirerek VXLAN olmayan sunucuların VXLAN ağında bulunan sanal makineler ile iletişimini sağlar.

Paylaşımlı kenar ve işlem kümesi içerisinde tüm NSX Edge örneklerini ve evrensel dağıtık mantıksal yönlendirici örneklerini (Universal Distributed Logical Router) barındırır. Bu sayede ağ yöneticisi ortamı daha güvenli ve merkezi olarak yönetebilir.

Tablo 62 - VXLAN Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-AĞ-015 | VXLAN ile sanal uygulama ve misafir ağlarının kullanımı için NSX for vSphere kullanılacaktır. | Her misafir için ağ yapılandırmasının basitleştirilmesi ve merkezi sanal ağ yönetimi için VXLAN kullanımı öngörülmektedir. | NSX bileşenlerinin kullanımı için ek işlem ve veri depolama kaynakları gerekmektedir. NSX için ek eğitim de gerekebilir. |
| SDDC-SA-AĞ-016 | VXLAN ile birlikte NSX Edge Geçitleri, UDLR (Universal Distributed Logical Router) ve DLR (Distributed Logical router) kullanılarak müşteri/misafir ağ kabiliyetleri sağlanacaktır. | Veri merkezleri arasında izole, birden çok misafir yayın alanı oluşturularak fiziksel sınırları geren esnek mantıksal ağlar oluşturulacaktır. | Transport ağları ve 1600 byte’dan büyük MTU’lar erişilebilirlik çeperinde yapılandırılmalıdır. |
| SDDC-SA-AĞ-017 | VXLAN ile birlikte NSX Edge Geçitleri, UDLR (Universal Distributed Logical Router) kullanılarak yönetim uygulama ağı kabiliyetleri sağlanacaktır. | Yönetim podu için de ağ sanallaştırmanın özelliklerinden faydalanılacaktır. | Yönetim kümesi için de NSX kurulu ve yapılandırılması gerekir. |

* + - 1. vMotion TCP/IP Yığını

vMotion TCP/IP yığını kullanarak trafik izole edilmelidir ve vMotion trafiği için adanmış bir ağ geçiti adresi kullanılmalıdır. Ayrı bir TCP/IP yığını kullanarak vMotion ve soğuk göç trafiği ağ topolojisine ve organizasyonun gereksinimlerine göre yönetilebilir. Çalışan veya çalışmayan sanal makinelerin göç trafiğini varsayılan sanallaştırma sunucusu ağ geçitinden farklı bir ağ geçiti adresine yönlendirmek gerekir. Ayrı bir yığın kullanarak ayrı bir set tampon ve soket kullanılmış olunur. Pek çok genel TCP/IP yığını kullanan özellik için gereken yönlendirme tablosundan farklı bir tablo kullanılacağından çakışmalar önlenmiş olur. Trafik izole edildiğinden güvenlik açısından da faydalıdır.

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-AĞ-018 | vMotion trafiği için vMotion TCP/IP yığını kullanılacaktır. | Ayrı bir TCP/IP yığını kullanıldığında farklı bir ağ geçiti kullanılacak ve bu sayede vMotion trafiği Layer 3 ağ üzerinden gitmesi mümkün olacaktır. | vMotion TCP/IP seçimi vDS VMkernel oluşturma sihirbazı üzerinde mevcut değildir. Bu yüzden VMkernel adaptörü her sanallaştırma sunucu üzerinde direk olarak oluşturulmalıdır. |

* + 1. NSX Tasarımı

Bu tasarımda yazılım tanımlı ağ VMware NSX for vSphere kullanılarak uygulanmaktadır. NSX kullanılarak sanallaştırma işlem ve veri depolama için ne sağlıyorsa ağ için de aynı yetenekler sağlanmaktadır.

Nasıl sunucu sanallaştırma programatik olarak yazılım tabanlı sanal makinelerin snapshot’larını oluşturup, silip yeniden yükleyebiliyorsa NSX’de programatik olarak yazılım tabanlı sanal ağların snapshot’larını oluşturabilir, silebilir ve yeniden yükleyebilir. Sonuç ise sadece veri merkezi yöneticilerine kat be kat fazla çeviklik ve ekonomik avantaj sağlamak değil aynı zamanda altyapı üzerinde aşırı kolay bir operasyonel model sunan, ağ tasarımına dönüştürücü bir yaklaşımdır. NSX herhangi bir mevcut IP ağına (geleneksel ağ modeli ve yeni jenerasyon fabrikler) kesintisiz olarak uygulanabilmektedir.

Sistem yöneticileri yeni iş yükleri provizyonladıklarında ağ yönetimi en çok zaman tüketen iş kalemlerinden birisi haline gelmektedir. Harcanan zamanın en büyük kısmı fiziksel alt yapı bünyesindeki bileşenlerin yapılandırılması ve yapılan değişikliklerin aynı alt yapıyı kullanan diğer cihazların etkilenmediğinin kontrolü esnasında gerçekleşmektedir.

Ağları önceden provizyonlanması ve yapılandırılmaları ihtiyacı, hızın, çevikliğin ve esnekliğin kritik gereksinimler olduğu bulut konumlandırmalarında en büyük engeldir. Önceden provizyonlanmış fiziksel ağlar, sanal ağların hızlı oluşturulmasına ve bu sanal ağları kullanacak iş yüklerinin hızlı konumlandırılmasına imkân vermektedir. İhtiyaç duyulan fiziksel ağ alt yapısı iş yükünün konumlandırılacağı sanallaştırma sunucularında mevcut ise bu yöntem iyi bir şekilde çalışır. Fakat sanallaştırma sunucularına bağlı böyle bir ağ mevcut değilse uygun başka bir sanallaştırma sunucusu bulunması gerekir.

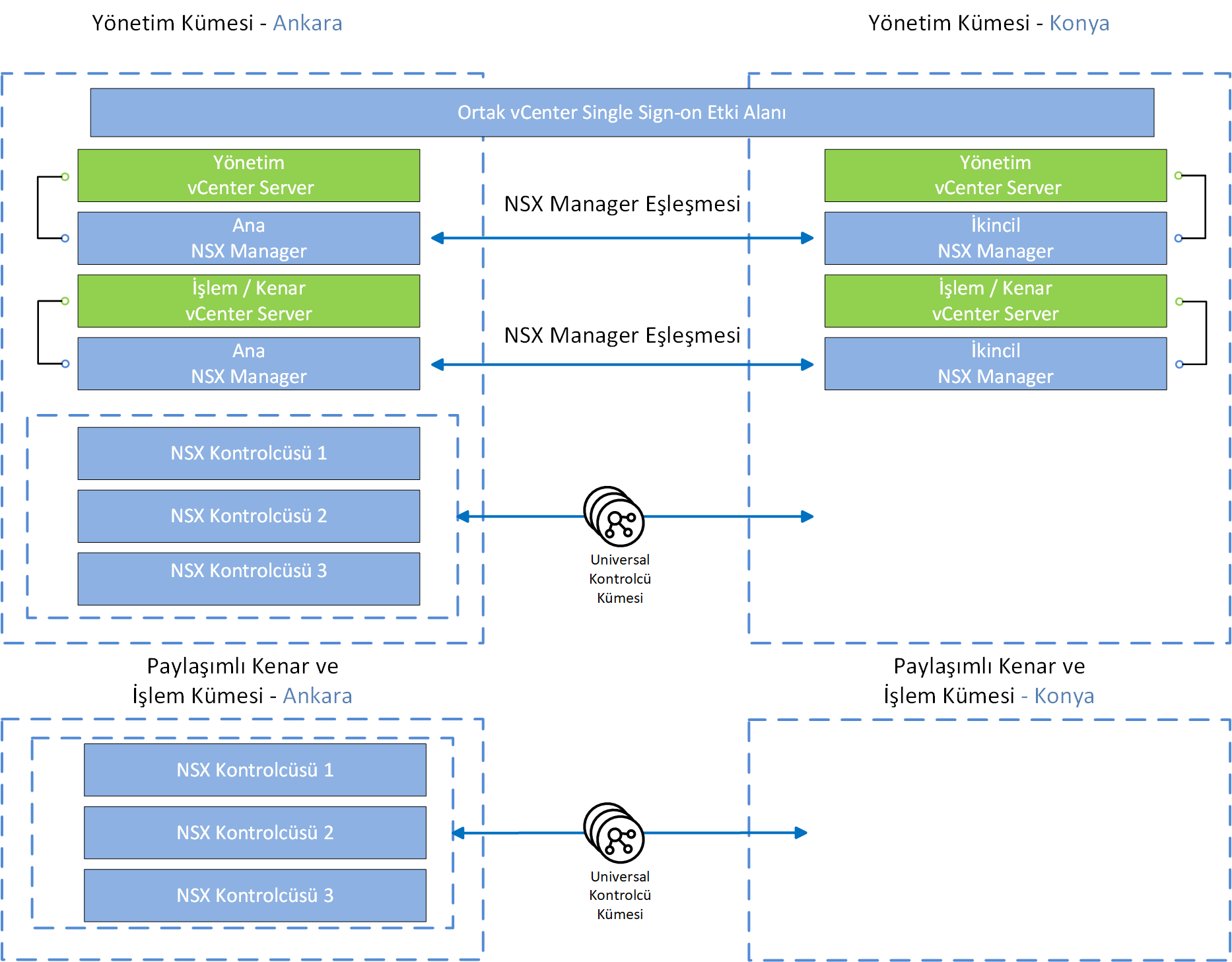
Bu darboğazı aşabilmek için sanal ağların fiziksel karşılıklarından ayrılmaları gerekmektedir. Bunun gerçekleştirebilmek için sanal ortamda çalışacak iş yüklerinin gerektirdiği fiziksel ağların özelliklerini programatik olarak yeniden oluşturmak gerekmektedir. Ağ sanallaştırma fiziksel ağları etkileyip değiştirmeden istenen sanal ağların oluşturulabilmesine imkân verdiğinden istenen ağların hızlıca provizyonlanması mümkün olabilmektedir.

* + - 1. NSX for vSphere Tasarımı

Her bir NSX oluşumu bir vCenter Server oluşumuna bağlıdır. Her bir bölge başına iki vCenter Server konumlandırma tasarım kararı (SDDC-SA-VC-001) bölge başına iki ayrı NSX oluşumu konumlandırmasını gerektirmektedir.

Tablo 63 - NSX Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-SDN-001 | Her bölge için iki ayrı NSX oluşumu yapılandırılacaktır. Bir oluşum yönetim vCenter Server’ına bağlı iken diğeri de işlem vCenter Server’ına bağlanacaktır. | Yük dengeleme ve güvenlik duvarı gibi NSX tarafından sunulan yazılım tanımlı ağ (SDN) özellikleri bulut yönetim platformu operasyonları için kritiktir. Yönetim yığını da bu kabiliyetlere ihtiyaç duymaktadır. | Birden fazla NSX oluşumunu ayrı olarak kurup yapılandırmak gerekir. |
| SDDC-SA-SDN-002 | Hem yönetim hem de işlem iş yükleri için NSX manager’lar bölgeler arasında birincil ve ikincil olarak yapılandırarak eşleştirilecektir. | NSX ağların mantıksal sınırlarını ve güvenlik hizmetlerini bölgeler arasında uzatabilmektedir. Sonuç olarak iş yükleri ağ üzerinde herhangi bir yapılandırma değişikliğine gitmeden canlı olarak göç ettirilebilir veya yük devredilebilir. | NSX manager oluşumları 8 adede kadar eşleştirilebilmektedir. |



Şekil 6 - NSX Mimarisi

* + - 1. NSX Bileşenleri

Bu bölümde çözüm içeresinde bulunan bileşenler ve ağ sanallaştırma ile olan ilgileri verilecektir.

* + - 1. Tüketim Katmanı

Bulut yönetim platformu (cloud management platform, CMP) NSX REST API ve vSphere Web Client kullanarak vRealize Automation tarafından temsil edilen NSX for vSphere’i tüketmektedir.

* + - 1. Bulut Yönetim Platformu

vRealize Automation yazılımı CMP adına NSX’i kullanır. NSX bir hizmet portalı aracılığıyla sanal ağların self servis provizyonlanmasını ve ilgili özelliklerin kullanılmasını sağlar.

* + - 1. Uygulama Programlama Ara Yüzü (API)

NSX, REST API aracılığıyla kuvvetli bir yönetim ara yüzü sağlar. Bir istemci bir nesneyi HTTP GET isteğini nesnenin kaynak URL’sine yönlendirerek okuyabilir. Bir istemci bir nesneye yeni ya da değiştirilmiş XML dokümanı eklenmiş bir HTTP PUT ya da POST isteği yaparak yazabilir (oluşturabilir ya da değiştirebilir). Son olarak bir istemci HTTP DELETE isteği yaparak bir nesneyi silebilir.

* + - 1. vSphere Web Client

NSX manager bileşeni vSphere Web Client için bir ağ yönetimi ve güvenliği eklentisi sağlamaktadır. Bu eklenti yeterli erişim hakkı bulunan kullanıcılar için sanal ağlar ile ilgili operasyonlar yapmasını sağlar.

Tablo 64 - Tüketim Yöntemleri Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-SDN-003 | Paylaşımlı kenar ve işlem kümeleri oluşumlarında son kullanıcı erişimi vRealize Automation hizmetleri ile sağlanacaktır. Yöneticiler vSphere Web Client ve NSX REST API’sini kullanabilirler. | vRealize Automation hizmetleri müşterinin erişebileceği bir portal tarafından sunulmaktadır. vSphere Web Client NSX’i eklenti aracılığıyla kullanmaktadır. REST API’si tekrarlanan aksiyonlar ve operasyonlar iin betikler sağlama potansiyeline sahiptir. | Müşteriler genellikle NSX ile vRealize Automation portalı ile etkileşime gireceklerdir. Yöneticiler NSX’i web client ya da REST API ile kullanacaklardır. |
| SDDC-SA-SDN-004 | Yönetim kümesi için NSX oluşumu sadece sağlayıcı personeli için Web Client ile ya da API ile sağlanacaktır. | Alt yapı bileşenlerinin misafir ya da sağlayıcı personeli olmayanlar tarafından değiştirilemeyeceğinden emin olunması gerekir. | Misafirlerin yönetim yığını iş yüklerine erişimler yoktur. |

* + - 1. NSX Manager

NSX manager, NSX for vSphere için merkezi bir yönetim düzlemi sağlar ve vCenter Server iş yükleri ile bire bir ilişkiye sahiptir. NSX manager aşağıdaki fonksiyonları yerine getirir:

* Yapılandırmalar için tek bir nokta sağlar ve vSphere ortamı içerisindeki NSX için REST API giriş noktasını oluşturur.
* NSX kontrolcüsü kümelerini, Edge dağıtık yönlendiricilerini ve Edge hizmet geçitlerini konumlandırır.
* UWA (User World Agent) aracılığıyla VXLAN, dağıtık yönlendirme ve güvenlik duvarı çekirdek modüllerini kurarak, ESXi sanallaştırma sunucularını hazırlar.
* NSX kontrolcü kümeleri ile REST üzerinden, sanallaştırma sunucuları ile RabbitMQ mesaj yolu üzerinden iletişim kurar.
* NSX kontrolcü oluşumları ve ESXi sanallaştırma sunucuları için sertifikalar üreterek kontrol düzlemi iletişimlerini karşılıklı kimlik doğrulama ile güvenli hale getirir.
  + - 1. NSX Kontrolcüsü

NSX kontrolcüsü aşağıdaki fonksiyonları gerçekleştirir:

* VXLAN ve mantıksal yönlendirme bilgilerini ESXi sanallaştırma sunucularına dağıtan kontrol düzlemini sağlar.
* Yüksek erişilebilirlik ve dağıtıklaştırmak için gerekli kümeleştirilmiş birimleri barındırır.
* Yedeklilik adına küme birimleri arasında ağ bilgilerine paylaştırır.
* Fiziksel ağ üzerindeki VXLAN Layer 3 multicast gerekliliğini ortadan kaldırır.
* VXLAN ağlarında yayım trafiğinin ARP baskılamasını sağlar.

Tablo 65 - NSX Kontrolcüsü Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-SDN-005 | Yüksek erişilebilirlik ve ölçeklenebilirlik sağlayabilmek için NSX kontrolcüsü Evrensel Küme kipinde ve üç üye ile kurulacaktır. Bu birimler ana NSX Manager oluşumu aracılığıyla konumlandırılacaktır. | NSX Kontrolcülerinin yüksek erişilebilirliği fiziksel sunucuların arızalanması durumunda kesinti olmasını engeller. | İkincil NSX Manager’ı kontrolcü konumlandırmayacaktır.  Ana NSX manager’ın kontrolcüleri ikincil kaynakları da yöneteceklerdir. |

* + - 1. NSX Sanal Anahtarı

NSX veri düzlemi NSX sanal anahtarlarından oluşur. Bu sanal anahtar diğer hizmetleri verebilmek için bulunan ekstra bileşenlere sahip vSphere Distributed Switch’e dayanır. Bu eklenti bileşenler hipervizör çekirdeği içinde çalışan ve dağıtık mantıksal yönlendirici (DLR) ve dağıtık güvenlik duvarı ve VXLAN kabiliyetlerinden oluşan çekirdek modüllerinden (VIB) oluşur.

NSX sanal anahtarı fiziksel ağı soyutlar ve hipervizör seviyesinde erişim seviyesi anahtarlama yapar. Ağ sanallaştırmasının temelinde bulunu çünkü VLAN’la gibi fiziksel yapılardan bağımsız mantıksal ağları etkinleştirir. NSX sanal anahtarın kullanımının çeşitli faydaları vardır:

* Örtüşme (overlay) ağını ve ağ yapılandırmalarını merkezileştirir.
* Hipervizörlerin büyük sayılara ölçeklenebilmelerini kolaylaştırır.
* NSX sanal anahtarı VDS tabanlı olduğundan, trafik yönetimi, izlemesi ve sorun gidermesi için port aynalama, NetFlow/IPFIX, yapılandırma yedeklenmesi, ağ sağılığı kontrolü vb. geniş bir araç koleksiyonu mevcuttur.
  + - 1. Mantıksal Anahtarlama

NSX mantıksal anahtarları misafir sanal makinelerin bağlandığı mantıksal olarak soyutlanmış parçaları oluştururlar. Tek bir mantıksal anahtar özgün bir VXLAN parçasına eşlenmiştir ve transport alanı içerisindeki ESXi sanallaştırma sunucularına dağıtılmış vaziyettedir. Mantıksal anahtar hipervizör anahtarlama işlerini hat hızında, VLAN yayılması veya spanning tree sınırlamalarına takılmadan gerçekleştirir.

* + - 1. Dağıtık Mantıksal Yönlendirici (Distributed Logical Router, DLR)

NSX dağıtık mantıksal yönlendiricisi sanallaştırılmış uzayda iletme (VXLAN üzerinde bulunan ya da VLAN destekli port gruplarına bağlı VM’ler arası) için optimize edilmiştir. DLR aşağıdaki özelliklere sahiptir:

* Yüksek performanslı, düşük ek yük getiren tek sıçramada yönlendirme.
* Sanallaştırma sunucu sayısı ile birlikte ölçeklenir.
* Her bir DLR üzerinde 1000 taneye kadar mantıksal ara yüz (Logical Interface, LIF)
  + - 1. Dağıtık Mantıksal Yönlendirici Kontrol Sanal Makinesi

Dağıtık mantıksal yönlendirici kontrol sanal makinesi, MSX manager ve NSX kontrolcüsü kümesi arasındaki iletişimi UWA (User World Agent) aracılığıyla yapan, yönlendirme sürecinin kontrol düzlem bileşenidir. NSX manager mantıksal ara yüze bilgilerini kontrol sanal makinesine ve NSX kontrol kümesine gönderir ve kontrol sanal makinesi de yönlendirme güncellemelerini NSX kontrolcü kümesine gönderir.

* + - 1. Kullanıcı Dünyası Ajanı (User World Agent, UWA)

UWA NSX manager’dan bilgi toplayan ve ESXİ sanallaştırma sunucuları ve NSX kontrolcüsü arasında iletişimi sağlayan bir TCP (SSL) istemcidir.

* + - 1. VXLAN Tüneli Bitiş Noktası (Tunnel Endpoint)

VXLAN tünel bitiş noktaları (VTEP) NSX için hazır olan ESXi sanallaştırma sunucularının bağlı oldukları vSphere Distributed Switch içinde somutlaştırılır. VTEP’ler VXLAN trafiğini UDP paketleri içerisinde sarmalamak ve geri açmaktan sorumludurlar. VTEP’ler IP adreslerine sahip bir ya da birden çok VMkernel portları şeklinde ortaya çıkarlar ve diğer VTEP’ler ile paket değiş tokuşu yapmak için ve IGMP aracılığıyla IP multicast grupları oluşturmak için kullanılırlar. Birden fazla VTEP kullanılacaksa bir birleştirme yöntemi belirlenmelidir.

* + - 1. Kenar Hizmetler Geçidi (Edge Services Gateway, ESG)

Kenar hizmetleri geçitlerinin ana fonksiyonu kuzey güney iletişimini sağlamakla beraber aynı zamanda Layer 2, Layer 3, çeper güvenlik duvarı, yük dengeleme ve SSL-VPN, DHCP-relay gibi diğer hizmetler için destek sağlamaktır.

* + - 1. Dağıtık Güvenlik Duvarı (Distributed Firewall)

NSX içerisinde dağıtık güvenlik duvarı olarak adlandırılan bir kernel seviyesi güvenlik duvarı barındırır. Fiziksel cihazlar üzerinde dar boğazlar yaratmadan yüksek bir şekilde ölçeklenebilir güvenlik duvarı uygulaması sağlanır. Dağıtık güvenlik duvarının minimal CPU gereksinimleri vardır ve hat hızında işlem yapabilir.

* + - 1. Mantıksal Yük Dengeleyici (Logical Load Balancer)

NSX mantıksal yük dengeleyici Layer 7 seviyesine kadar yük dengeleme hizmetleri ile trafiğin birden çok sunuculara dağıtarak optimal kaynak kullanımı ve erişilebilirlik sağlar. Mantıksal yük dengeleyici NSX kenar hizmetler geçidi (ESG) tarafından verilen bir servistir.

* + - 1. NSX Gereksinimleri

NSX gereksinimlerinin hem fiziksel hem de sanal ağlar üzerinde etkileri bulunmaktadır. Fiziksel gereksinimler VLAN trafiğini taşıyan ağların MTU büyüklüklerini, dinamik yönlendirme desteğini, NTP sunucusu senkronizasyonunu ve ileri ve geri yönde DNS çözümlemelerini belirler.

|  |  |
| --- | --- |
| Gereksinimler | Yorumlar |
| VXLAN trafiği taşıyan tüm ağların MTU büyüklükleri 1600 ya da daha büyük olmalıdır | VXLAN paketleri parçalanmış olamazlar. MTU büyüklüğü sarmalamadan gelen ek yükü destekleyecek kadar büyük olmalıdır. Bu tasarımda VXLAN trafiği için MTU büyüklüğü 9000 olan Jumbo Frames kullanılmaktadır. |
| Hibrid replikasyon kipi için IGMP snooping ESXi sunucuların bağlı olduğu tüm Layer 2 anahtarlarda açılmalıdır. IGMP querier bağlı bir yönlendirici de ya da Layer 3 anahtarda açılmalıdır. | Hibrid replikasyon kipi yayım, bilinmeyen tekli yayım ve çoklu yayım trafiği için tavsiye edilen replikasyon kipidir. Geleneksel protokol bağımsız çoklu yayım (PIM) gereksinimi kaldırılmıştır. |
| Yukarı yönlü Layer 3 veri veri merkezi anahtarlarından dinamik yönlendirme desteği etkinleştirilmelidir. | NSX tarafından desteklenen dinamik yönlendirme protokollerinin veri yukarı yönlü veri merkezi anahtarları üstünde etkileştirilmesi komşu ESG’ler arasında dinamik yönlendirme yapılabilmesini sağlar. |
| NTP sunucuları erişilebilir olmalıdır. | NSX manager’ın diğer vSphere ortamı ile senkron olabilmesi için ntp sunucularına ihtiyacı vardır. Zaman kayması kimlik onaylanmasında sorunlara yol açabilir. NSX manager’ın vCenter SSO ile senkronize olması gerekir. |
| Tüm yönetim VM’leri için ileri ve geri yönlü DNS çözümlemeleri tanımlı olmalıdır. | NSX kontrol birimleri DNS girişlerine ihtiyaç duymazlar. |

* + - 1. NSX Bileşen Özellikleri

Aşağıdaki tabloda NSX çözümü için gereken bileşenler ve onların kurulumları ve çalıştırılmaları için gereksinimleri verilmiştir. Kaynak miktarları belirlenirken NSX’i destekleyecek işlem ve veri depolama gereksinimleri de göz önüne alınmıştır.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| VM | vCPU | Hafıza | Veri Depolama | Oluşum Yığını Başına Adet |
| NSX Manager | 4 | 16 GB | 60 GB | 1 |
| NSX Controller | 4 | 4 GB | 20 GB | 3 |
| NSX ESG | 1 (Compact)  2 (Large)  4 (Quad Large)  6 (X-Large) | 512 MB (Compact)  1 GB (Large)  1 GB (Quad Large)  8 GB (X-Large) | 512 MB  512 MB  512 MB  4.5 GB (X-Large)  (+4 GB swap ile) | Opsiyonel bileşen, NSX ESG kullanım durumuna göre değişir. |
| DLR Kontrol VM | 1 | 512 MB | 512 MB | Opsiyonel bileşen, Genellikler HA çifti başına 2 adet. |
| Guest Introspection | 2 | 1 GB | 4 GB | Opsiyonel bileşen,ESXi sunucu başına 1 adet. |
| NSX Data Security | 1 | 512 MB | 6 GB | Opsiyonel bileşen,ESXi sunucu başına 1 adet. |

* + - 1. NSX Kenar Hizmetler Geçiti Boyutlandırma

Quad Large modeli yüksek performanslı güvenlik duvarı yetenekleri için, X-Large modeli ise hem yüksek performanslı yük dengeleme hem de yönlendirme için uygundur.

NSX kenar hizmetler geçitlerinin boyutlarını kesintisiz bir güncelleme süreci ile artırmak mümkündür. Tavsiye olunan Large model ile başlanması ve gerektiğinde ölçeklenecek boyuta göre artırıma gidilmesidir. Large NSX kenar hizmetler geçiti ortalama bir güvenlik duvarı performansı için yeterlidir ama ileride ayrıntılanacağı gibi güvenlik duvarı fonksiyonlarının çoğunluğunu geçit gerçekleştirmeyecektir.

Tablo 66 - NSX Kenar Hizmet Geçiti Boyutlandırma Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-SDN-006 | NSX hizmet geçitleri için Large modeli kullanılacaktır. | Large boyutu arıza halinde bile tüm performans gereksinimlerini karşılamaktadır. Daha büyük bir boyut istenen performansı karşılayabilir fakat bunun karşılığında ekstra kaynak gerekecektir. | Yoktur. |

* + - 1. Kavramsal Ağ Sanallaştırma Tasarımı

Kavramsal tasarım, ağ sanallaştırma tasarımının anlaşılmasını kolaylaştırmak amacıyla verilmiştir. Kavramsal tasarımda çeper güvenlik duvarı, sağlayıcı mantıksal yönlendirici ve NSX mantıksal yönlendiricisi bulunmaktadır. Bunlarla birlikte harici ağ, iç misafir ağı ve iç yönetim ağı yer almaktadır.



Şekil 7 - Kavramsal Misafir Ağı Genel Bakış

Kavramsal tasarım aşağıdaki anahtar bileşenlere sahiptir:

* Harici Ağlar: Harici ağlara gönderilen ve oradan gelen tüm trafik çeper güvenlik duvarından geçmektedir. Ana harici ağ Internet’dir.
* Çeper Güvenlik Duvarı: Fiziksel güvenlik duvarı veri merkezinin çeperinde bulunmaktadır. Her bir misafir harici trafiğin filtrelenmesi için ya tam bir oluşum ya da oluşum bölümü alırlar.
* Sağlayıcı Mantıksal Yönlendirici (PLR): PLR çeper güvenlik duvarı arkasında yer alır ve misafir iş yükleri için giriş ve çıkış yapan kuzey güney yönlü idare eder.
* NSX Dağıtık Mantıksal Yönlendirici (DLR): Mantıksal yönlendirici VM’ler, VXLAN port grupları ya da VLAN destekli port grupları olan sanal uzaydaki iletimler için optimize edilmiştir.
* İç Misafir için olmayan Ağ: Çeper güvenlik duvarı arkasında olan fakat PLR arkasında olmayan, tek bir yönetim ağıdır. Müşterilerin misafir ortamları yönetmelerini sağlar.
* İç Misafir Ağı: Ana misafir iş yükleri için olan ağdır. Bu ağlar PLR arkasında duran DLR’ye bağlı olan ağdır. Bu ağlar VXLAN tabanlı NSX mantıksal anahtarlar şeklinde ortaya çıkarlar. Misafir sanal makine iş yükleri direk olarak bu ağlara bağlanırlar.
  + - 1. NSX for vSphere için Küme Tasarımı

vSphere tasarımını takip eden NSX tasarımı da her bir bölge için bir yönetim yığınından ve bir işlem/kenar yığınından oluşmaktadır.

Yönetim yığınında bulunan sanallaştırma sunucuları NSX for vSphere için hazırlanırlar. Yönetim yığını aşağıdaki bileşenlerden oluşur:

* Her iki yığın (yönetim ve işlem/kenar yığınları) için NSX Manager oluşumları.
* Yönetim yığını için NSX Kontrolcü kümesi.
* Yönetim yığını içi NSX ESG ve DLR kontrol sanal makineleri

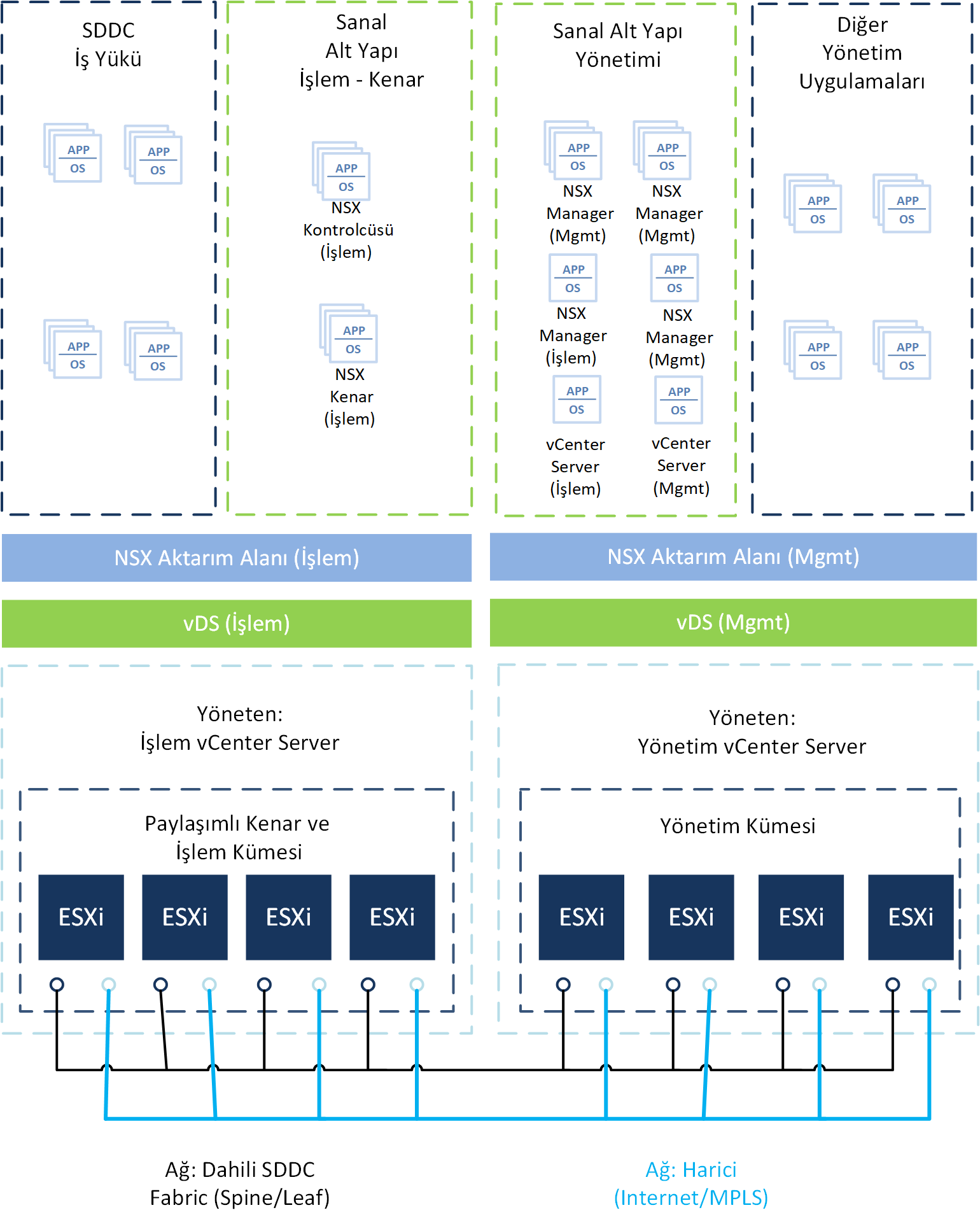
İşlem/kenar yığınında bulunan sanallaştırma sunucuları NSX for vSphere için hazırlanırlar. İşlem/kenar yığını aşağıdaki bileşenlerden oluşur:

* İşlem yığını için NSX Kontrolcüsü
* Veri merkezi içerisindeki kuzey güney trafiği idare etmek için adanmış, işlem yığınının tüm NSX Kenar hizmet geçitleri ve DLR kontrol sanal makineleri.

Tablo 67 - vSphere Küme Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-SDN-007 | İşlem yığını için adanmış bir kenar kümesi kullanılmayacaktır. | Yapılandırmayı basitleştirir ve ilk kurulumda gereken sanallaştırma sunucusu sayısını azaltır. | İşlem yığınının NSX Kontrolcüleri, NSX Kenar hizmet geçitleri ve DLR kontrol sanal makineleri paylaşımlı kenar ve işlem kümesinde konumlandırılırlar |
| SDDC-SA-SDN-008 | Yönetim yığını için adanmış bir kenar kümesi kullanılmayacaktır. | Desteklenen yönetim uygulamalarının sayısı yönetim yığınında adanmış bir kenar kümesini gerektirmemektedir. | Yönetim yığınının NSX kontrolcüleri, NSX kenar hizmet geçitleri ve DLR kontrol sanal makineleri yönetim kümesinde konumlandırılırlar. |
| SDDC-SA-SDN-009 | Her bir yığın için olan NSX bileşenleri için vSphere Distributed Resource Scheduler (DRS) anti affinity kuralı tanımlanacaktır. | DRS kullanımı kontrolcülerin aynı ESXi sanallaştırma sunucusu üzerinde çalışmalarına engel olarak arıza durumunda yüksek erişilebilirliğin risk altında kalmasını engeller. | Anti affinity kuralı tanımlanması için ek yapılandırma yapılması gerekir. |

NSX’in mantıksal tasarımı vCenter Server kümelerini dikkate alır ve her bir NSX bileşeninin nereye yerleştirileceğini tanımlar.



Şekil 8 - Yönetim ve Paylaşımlı Kenar ve İşlem Podu

* + - 1. NSX Bileşenleri için Yüksek Erişilebilirlik

Her iki yığının NSX manager oluşumları da yönetim kümesi üzerinde çalışırlar. vSphere HA NSX manager sanal makinelerini arıza durumunda başka bir sanallaştırma sunucusunda tekrar başlatarak NSX manager oluşumlarını korurlar.

Yönetim yığınının NSX kontrolcü birimleri yönetim kümesinin üzerinde çalışırlar. İşlem yığınının NSX kontrolcü birimleri paylaşımlı kenar ve işlem kümesinde çalışırlar. Her iki küme de vSphere DRS kuralları NSX kontrolcü birimlerinin aynı sanallaştırma sunucusu üzerinde çalışmalarına engel olurlar.

Yönetim ve kontrol düzlemlerinde oluşabilecek kesintilerde veri düzlemi aktif olarak çalışmaya devam eder ama bu düzlemler tekrar erişilebilir hale gelene kadar sanal ağların provizyonlanması ve değiştirilmeleri mümkün değildir.

İşlem yığının NSX Kenar hizmet geçitleri ve DLR kontrolcü sanal makineleri paylaşımlı kenar ve işlem kümesinde yer alırlar. Yönetim yığınının NSX kenar hizmet geçitleri ve DLR kontrol sanal makineleri yönetim kümesinde çalışırlar.

Kuzey güney trafiği için konumlandırılan NSX kenar bileşenleri, saniyeler içerisinde yönlendirme yük devralmasını destekleyen eşit maliyet çoklu yol (equal-cost multi-path, ECMP) ile yapılandırılmışlardır. Yük dengelemesi için konumlandırılan NSX kenar bileşenleri NSX HA’yı kullanırlar. NSX HA, vSphere HA’dan daha hızlı olarak kurtarma sağlarlar çünkü NSX HA aktif pasif bir çift NSX kenar servisi kullanır. Varsayılan ayarlar ile pasif kenar cihazı 15 saniye içerisinde aktif hale geçer. Tüm NSX kenar ciharları aynı zamanda vSphere HA ile de korunurlar.

* + - 1. NSX Bileşenlerinin Ölçeklenebilmesi

NSX Manager oluşumları ile vCenter Server oluşumları arasında bire bir ilişki mevcuttur. Eğer yönetim yığını ya da işlem yığınından herhangi biri tek bir vCenter tarafından desteklenen limitleri aşarlarsa yeni bir vCenter Server oluşumu konumlandırılabilir ve dolayısı ile yeni bir NSX Manager oluşumu da konumlandırılır. vCenter Server limitlerine ulaşana kadar paylaşımlı kenar ve işlem ve sadece işlem kümeleri ekleyerek aktarım bölgeleri genişletilebilir. Her bir ESXi sanallaştırma sunucusu başına 100 DLR limit bulunmakla beraber böyle bir ortam DLR limitlerinden çok daha önce vCenter limitlerine ulaşmış olur.

* + - 1. vSphere Dağıtık Anahtar Uplink Yapılandırması

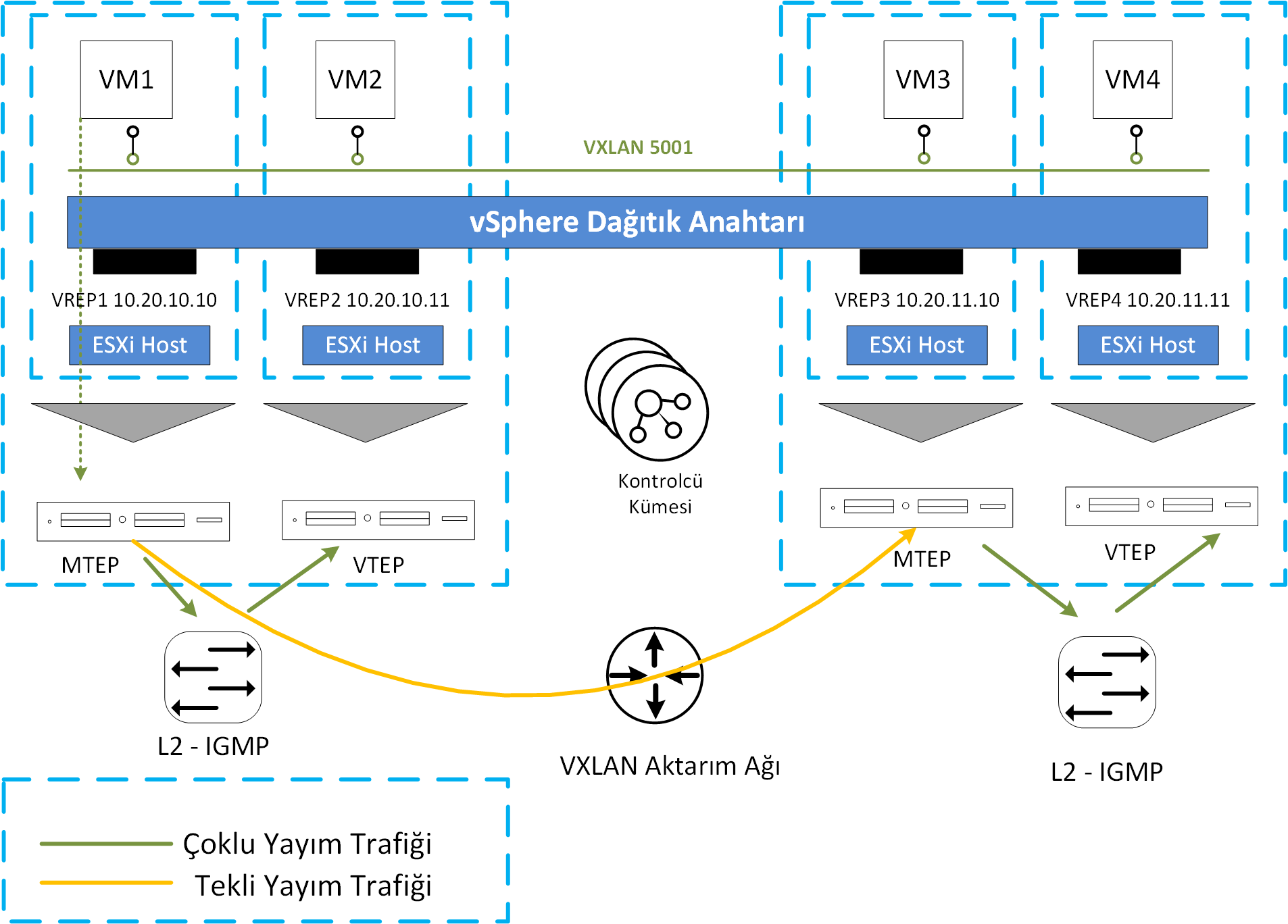
Her bir ESXi sanallaştırma sunucusu iki adet fiziksel 10Gb Ethernet adaptör kullanır. Bu adaptörler vSphere distributed switch uplinkleri ile ilişkilendirilirler. Her bir uplink farklı bir kabin üstü anahtara bağlanırlar. Böylece tek bir kabin üstü anahtar arızasının etkileri azaltılırken SDDC içine ve dışına iki ayrı yol sağlanmış olur.

Tablo 68 - VTEP Birleştirme ve Yük Devri Yapılandırma Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-SDN-010 | Birleştirme ve yük devri yapılandırması için VXLAN Tünel bitiş noktalarını (VTEP) Route based on SRC-ID olarak kullanılacaktır. | vDS’in her iki uplinkininde kullanılarak daha iyi bir bant genişliği kullanımı ve ağ yolları arızalarında daha hızlı kurtarma sağlanır. | Kabin üstü anahtarlar ve ESXi sanallaştırma sunucusu arasında LACP gibi link birleştirme yapılandırılmamalıdır. |

* + - 1. Mantıksal Anahtar Kontrol Düzlem Kipi Tasarımı

Kontrol düzlemi NSX’i fiziksel ağdan ayırır ve mantıksal anahtarlar içerisinde yayım, bilinmeyen tek noktaya yayım ve çoklu yayım (BUM) işlerine bakar. Kontrol düzlemi aktarım bölgesinin tepesinde yer alır ve bünyesinde oluşturulan tüm mantıksal anahtarlar intikal ederler. Üç ayrı seçenek mevcuttur: Çoklu yayım kipinde kontrol düzlemi fiziksel ağ üzerinde çoklu yayım IP’lerini kullanır. Çoklu yayım kipi sadece mevcut VXLAN konumlandırmalarından yükseltme yapılırken kullanılmalıdır. Bu kipte fiziksel ağ üzerinde PIM/IGMP yapılandırılması gerekir. Tek noktaya yayım kipinde kontrol düzlemi NSX kontrolcüleri tarafından gerçekleştirilir ve tüm replikasyonlar yerel olarak sanallaştırma sunucusunda gerçekleşir. Hibrid kipi ise subnet için yerel trafik replikasyonunun fiziksel ağa devredildiği, tek noktaya yayım kipinin optimize halidir. Hibrid kip ilk sıçrama anahtarında IGMP snooping etkinleştirilmesini ve her VTEP subnetinde bir IGMP sorgulayıcısı olmasına ihtiyaç duyar. Hibrid kipte PIM’e gerek yoktur.



Şekil 9 - Hibrid Kipte Mantıksal Anahtar Kontrol Düzlemi

Bu tasarımda kontrol düzlemi replikasyonu için hibrid kip kullanılmaktadır.

Tablo 69 - Mantıksal Anahtar Kontrol Düzlemi Kipi Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-SDN-011 | Kontrol düzlemi replikasyonu için hibrid kip kullanılacaktır. | Çoklu yayım işlemlerini fiziksel ağa devretmek ortam büyüdükçe VTEP’ler üzerindeki oluşacak baskıyı azaltır. Büyük ortamlar için hibrid kipte tekli yayım kipine tercih edilir. Çoklu yayım kipi isi sadece yükseltmelerde kullanılır. | Fiziksel ToR anahtarlarda IGMP snooping etkinleştirilmeli ve IGMP sorgulayıcı erişilebilir olmalıdır. |

* + - 1. Aktarım Bölgesi Tasarımı

Bir aktarım bölgesi, bir VXLAN üst ağın kapsamı olarak tanımlanır ve bir vCenter Server etki alanındaki bir ya da birden çok kümeyi kapsayabilir. NSX içerisinde bir ya da birden çok aktarım bölgesi tanımlanabilir. Bir aktarım bölgesi güvenlik sınırı anlamına gelmemektedir.

Tablo 70 - Aktarım Alanı Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-SDN-012 | İşlem yığını için, tüm bölgeler arasında hareketlilik gerektiren iş yükleri için olan tüm paylaşımlı kenar ve işlem kümelerini ve işlem kümelerini kapsayan bir evrensel aktarım alanı kullanılacaktır. | Evrensel bir aktarım bölgesi ağların ve güvenlik politikalarının bölgeler arasında yayılmasını sağlar. Bu sayede uygulamaların vMotion ya da SRM ile bölgeler arasında kesintisiz geçişi mümkün olur. | vRealize Automation ikincil NSX Manager’a istek üzerine ağ nesneleri konumlandıramaz. 8 adete kadar NSX Manager eşleştirilebileceği unutulmamalıdır. Fazlası için yeni bir ana Manager ve aktarım bölgesi konumlandırılmalıdır. |
| SDDC-SA-SDN-013 | İşlem yığını için, her bir bölgede, vRealize Automation’nın istek üzerine ağlar konumlandırabilmesi için olan paylaşımlı kenar ve işlem kümelerini kapsayan bir global aktarım alanı kullanılacaktır. | İkincil roldeki bir NSX Manager evrensel nesneler konumlandıramaz. Tüm bölgelerin istek üzerine ağ nesneleri konumlandırabilmesi için global aktarım alanı gerekir. | Paylaşımlı kenar ve işlem ve işlem podları iki aktarım alanına sahiptir. |
| SDDC-SA-SDN-014 | Yönetim yığını için tüm yönetim kümelerini kapsayan tek bir evrensel aktarım alanı kullanılacaktır. | Evrensel bir aktarım bölgesi ağların ve güvenlik politikalarının bölgeler arasında yayılmasını sağlar. Bu sayede yönetim uygulamaların vMotion ya da SRM ile bölgeler arasında kesintisiz geçişi mümkün olur. | 8 adete kadar NSX Manager eşleştirilebileceği unutulmamalıdır. Fazlası için yeni bir ana Manager ve aktarım bölgesi konumlandırılmalıdır. |
| SDDC-SA-SDN-015 | Yönetim yığını aktarım alanında kontrolcü ayrık operasyon (Controller Disconnected Operation, CDO) kipi kullanılacaktır. | NSX kontrolcülerinin ESXi sanallaştırma sunucuları veri düzlemi güncellemeleri ile iletişime geçemediği durumlarda VNI’ların aktif olması hala gerçekleşebilir. | CDO kipinin etkinleştirilmesi kontrol kümesi kapalı iken hipervizörler fazladan iş yükü getirebilir. |
| SDDC-SA-SDN-016 | Paylaşımlı kenar ve işlem evrensel aktarım alanında kontrolcü ayrık operasyon (Controller Disconnected Operation, CDO) kipi kullanılacaktır | NSX kontrolcülerinin ESXi sanallaştırma sunucuları veri düzlemi güncellemeleri ile iletişime geçemediği durumlarda VNI’ların aktif olması hala gerçekleşebilir. | CDO kipinin etkinleştirilmesi kontrol kümesi kapalı iken hipervizörler fazladan iş yükü getirebilir. Aynı dağıtık anahtar üzerinde paylaşımlı kenar ve işlem podunda ve gelecekteki işlem podlarında iki aktarım alanı olduğundan CDO kipi sadece bir aktarım alanında etkinleşebilir. |

* + - 1. Yönlendirme Tasarımı

Yönlendirme tasarımı ortam içerisinde farklı seviyelerdeki yönlendirmeleri ele alır ve ölçeklenebilir bir yönlendirme çözümü sunabilmek için prensipler seti tanımlar.

* Kuzey-güney: Sağlayıcı mantıksal yönlendirici (PLR), sanal ağlar içindeki misafir ve yönetim uygulamalarına giren ve çıkan kuzey-güney trafiğini ele alır.
* Doğu-batı: PLR alt katmanındaki dahili doğu-batı yönlendirmesi uygulama iş yüklerini ele alır.

Tablo 71 - Yönlendirme Modeli Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-SDN-017 | Yönetim ve paylaşımlı kenar ve işlem kümeleri içindeki kuzey-güney yönlendirmeleri için NSX Kenar Hizmetleri Geçitleri ECMP yapılandırılarak konumlandırılacaktır. | Kuzey-güney trafiği için tavsiye edilen cihaz NSX ESG’dir. ECMP kullanılması SDDC’ye birden çok yoldan giriş ve çıkış sağlar. Bu da kenar hizmetleri geçitlerinin HA kipinden daha hızlı yük devri sağlar. | ECMP uplinkler için geleneksel HA ESG yapılandrırmasına kıyasla bir fazla olan, 2 adet VLAN gerektirir. |
| SDDC-SA-SDN-018 | Tüm bölgeler arasında doğu-batı yönlendirmesini sağlayabilmek için yönetim kümesinde tek bir NSX UDLR konumlandırılacaktır. | UDLR kullanımı kendisine bağlı birimler arasında sıçrama sayısını bire indirir. Bu da gecikme sürelerini azaltır ve performansı artırır. | UDLR 1000 ara yüz ile sınırlıdır. Bu limite ulaşıldığında yeni bir UDLR konumlandırılmalıdır. |
| SDDC-SA-SDN-019 | Tüm bölgeler arasında doğu-batı yönlendirmesini sağlayabilmek için paylaşımlı kenar ve işlem ve işlem kümelerinde tek bir NSX UDLR konumlandırılacaktır. | UDLR kullanımı kendisine bağlı birimler arasında sıçrama sayısını bire indirir. Bu da gecikme sürelerini azaltır ve performansı artırır. | UDLR 1000 ara yüz ile sınırlıdır. Bu limite ulaşıldığında yeni bir UDLR konumlandırılmalıdır. |
| SDDC-SA-SDN-020 | vRealize Automation tarafından istek anında üretilen ağ nesnelerini gerektiren iş yüklerinin doğu-batı yönlendirmesini sağlamak için paylaşımlı kenar ve işlem ve işlem kümeleri üzerinde DLR konumlandırılacaktır. | DLR kullanımı kendisine bağlı birimler arasında sıçrama sayısını bire indirir. Bu da gecikme sürelerini azaltır ve performansı artırır | DLR 1000 ara yüz ile sınırlıdır. Bu limite ulaşıldığında yeni bir UDLR konumlandırılmalıdır |
| SDDC-SA-SDN-021 | Tüm NSX UDLR’leri yerel ağ çıkışı seçeneği kapalı olarak konumlandırılacaktır. | Yerel ağ çıkışı seçeneği etkinleştirildiğinde içeri giren trafiğin de kontrolü de (örnek NAT kullanımı) gereklidir. Fayda sağlamadığı gibi yönetim zorluğu getirmektedir. | Tüm kuzey-güney trafiği A bölgesinden yönlendirilmektedir ta ki bu yollar erişilebilir olmayan kadar. Böyle bir durumda da tüm trafik dinamik olarak B bölgesine geçer. |
| SDDC-SA-SDN-022 | SDDC içerisinde dinamik yönlendirme protokolü olarak BGP kullanılacaktır. | OSPF yerine BGP kullanılması dinamik yönlendirme uygulanmasını kolaylaştırır. SDDC içerisinde OSPF 0 alanı planlayıp erişim tasarlamaya gerek yoktur. OSPF 0 alanı müşteri yapılandırmasına göre değişiklik gösterir. | BGP her bir ESG ve UDLR’nin yönlendirme bilgilerini değiş tokuş ettiği uzak yönlendirici ile yapılandırılmasını gerektirir |
| SDDC-SA-SDN-023 | Kuzey-güney yönlendirmesini sağlayan UDLR ve tüm ESG’ler arasında BGP Keep Alive zamanlayıcısı 1, Hold Down zamanlayıcısı 3 olarak ayarlanacaktır. | UDLR ve ECMP ESG’lerinin arasında Keep Alive ve Hold zamanlayıcı değerleri düşük tutularak arızalar daha hızlı tespit edilir ve yönlendirme tabloları daha sık güncellenir. | Bir ESXi sanallaştırma sunucusu kaynak açısınında sıkıntıya düşerse, üzerindeki ESG çalışır vaziyette olmasına rağmen kullanılamayabilir. |
| SDDC-SA-SDN-024 | Kuzey-güney yönlendirmesini sağlayan tüm ESG’ler ve kabin üstü anahtarlar arasında BGP Keep Alive zamanlayıcısı 4, Hold Down zamanlayıcısı 12 olarak ayarlanacaktır. | Bu değerler kabin üstü anahtarlar ve ESG’ler arasında arıza tespiti ile keep alive trafiğinin kabin üstü anahtarları yorması arasında makul bir denge sağlar. | Daha yüksek değerler kullanılması ölü bir yönlendiricinin tespit süresini uzatır ve bu süre boyunca yönlendirme tablosu bu yönlendiriciyi barındırdığından trafik bu müddet boyunca ölü yönlendiriciye gider. |
| SDDC-SA-SDN-025 | UDLR ve DLR arkasındaki subnetler için ECMP etkinleştirilmiş kenarlar üzerinde dinamik olarak öğrenilmiş yollardan daha yüksek maliyetli bir ya da daha fazla statik yol tanımlanacaktır. | UDLR ya da DLR kontrol sanal makinesi arızalanırsa yönlendirici komşuluğu kaybolur ve yukarı yönlü cihazlardan gelen yollar kaybolur. | ECMP kenar cihazların UDLR ya da DLR’ye giden statik yollar ile yapılandırılması gerekir. Eğer UDLR veya DLR arkasına yeni subnetler eklenirse ECMP kenarlarda güncellenmesi gerekir. |

* + - 1. Transit Ağ ve Dinamik Yönlendirme

Evrensel dinamik yönlendiriciler ve kenar geçitleri arasındaki trafiğe ve kenar geçitleri ve kabin üstü anahtarlar arasındaki trafiğe olanak sağlamak için atanmış ağlar gerekir. Bu ağlar yönlendirme tablolarının bilgilerinin değiş tokuşu ve transit trafiği taşımak için kullanılırlar.

Tablo 72 - Transit Ağı Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-SDN-026 | UDLR ve ESG’ler arasında transit ağ olarak kullanılmak üzere evrensel bir sanal anahtar oluşturulacaktır. ESG’ler kuzey-güney yönlendirmelerini sağlarken, UDLR ise işlem ve yönetim yığınları için doğu-batı yönlendirmelerini sağlayacaktır. | Evrensel sanal anahtar UDLR ve ESG’lerin tüm bölgeler arasında yönlendirme bilgilerinin değiş tokuş yapılabilmesine izin verir. | Sadece ana NSX Manager bu UDLR gibi evrensel nesneleri oluşturabilir ve yönetebilir. |
| SDDC-SA-SDN-027 | DLR ve ESG’ler arasında transit ağ olarak kullanılmak üzere global bir sanal anahtar oluşturulacaktır. ESG’ler kuzey-güney yönlendirmelerini sağlarken, DLR işlem yığınında doğu-batı yönlendirmelerini sağlayacaktır. | Global sanal anahtar DLR ve ESG’lerin tüm bölgeler arasında yönlendirme bilgilerinin değiş tokuş yapılabilmesine izin verir | Her bir bölge için transit ağ olarak kullanılacak bir global sanal anahtar gereklidir. |
| SDDC-SA-SDN-028 | Her bölgede iki adet VLAN oluşturulacaktır. Bu VLAN’lar kuzey-güney ESG’leri ile ToR Anahtarlar arasında ECMP’nin etkileştirilmesi için kullanılacaktır. | ESG’lerin birden çok eşit-maliyet yolları olacaktır ve daha dirençli ve ağ bant genişliğinin daha verimli kullanımını sağlayacaktır. | Ekstra VLAN’lar gerekmektedir. |

* + - 1. Güvenlik Duvarı Mantıksal Tasarımı

NSX dağıtık güvenlik duvarı, sanal ağlara bağlı tüm yönetim uygulamaları korumak için kullanılır. SDDC’yi güvenli kılmak için bireysel bileşenlerle sadece SDDC içerisindeki diğer çözümler ve onaylanmış yönetim IP’leri direkt olarak iletişime geçebilmelidir. Dışarıya bakan portaller yük dengeleyicisinin sanal IP’si (VIP) aracılığıyla erişilmelidir.

Tüm güvenlik duvarı kurallarının yönetimini tek bir noktadan olması tasarımı kolaylaştırır. Bireysel ESG’ler üzerindeki güvenlik duvarları tüm trafiğe müsaade etmek üzere yapılandırılmıştır. Bunun istisnası ise ECMP hizmetleri sağlayan ESG’lerdir. Bunlar üzerinde güvenlik duvarı etkin değildir.

Tablo 73 - Güvenlik Duvarı Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-SDN-029 | Tüm ESG’ler yük dengeleyici olarak konumlandırılacak ve güvenlik duvarı ayarları tüm trafiğe müsaade edecektir. | Erişime müsaade edilmesi veya engellenmesi dağıtık güvenlik duvarı tarafından sağlanmaktadır. Varsayılan güvenlik duvarı kuralı bu işi yapmak zorunda değildir. | Dağıtık güvenlik duvarı üzerinde yönetim uygulamalarının erişimleri için açık kurallar tanımlanması gerekir. |
| SDDC-SA-SDN-030 | ECMP kuzey-güney yönlendirici olarak konumlandırılan tüm ESG’lerde güvenlik duvarı devre dışı bırakılacaktır. | ESG’ler üzerinde ECMP kullanımı bir gereksinimdir. Güvenlik duvarı ayarlarını etkinleştirmek, tüm trafiği geçirmek için bile olsa münferit ağ erişimlerine yol açar. | NAT ve yük dengeleyici hizmetleri güvenlik duvarı devre dışı iken kullanılamaz. |
| SDDC-SA-SDN-031 | Yönetim kümesindeki idari ara yüzlere olan erişimin sınırlanması için dağıtık güvenlik duvarı yapılandırılacaktır. | Sadece yetkili yöneticilerin idari ara yüzlere erişim için sınırlama gereklidir. | Güvenlik duvarı kurallarının idamesi ekstra iş yükü doğurur. |

* + - 1. Yük Dengeleyici Tasarımı

ESG NSX for vSphere içerisinde yük dengeleyici görevini yerine getirir. ESG’nin farklı özelliklere sahip hem Layer 4 hem de Layer 7 yük dengeleyici motoru mevcuttur. Aşağıdaki tabloda bu özellikler gösterilmiştir:

|  |  |  |
| --- | --- | --- |
| Özellik | Layer 4 Motoru | Layer 7 Motoru |
| Protokoller | TCP | TCP, HTTP, HTTPS (SSL Pass Thru), HTTPS (SSL Offloading) |
| Yük Dengeleme Mekanizması | Round Robin, Source IP Hash, Least Connection | Round Robin, Source IP Hash, Least Connection, URI |
| Sağlık Kontrolü | TCP | TCP, HTTP (GET, OPTION, POST), HTTPS (GET, OPTION, POST) |
| Süreklilik (istemci bağlantılarının hep aynı arka sunucuya gitmesi) | TCP:Source IP | TCP:Source IP, MSRDP, HTTP: Source IP, Cookie, HTTPS: Source Ip, Cookie, ssl\_session\_id |
| Bağlantı Kısma | Yok | İstemci tarafı: Maksimum eş zamanlı bağlantılar, Saniyede maksimum yeni bağlantı, Sunucu tarafı: Maksiumum eş zamanlı bağlantılar |
| Yüksek Erişilebilirlik | Var | Var |
| İzleme | Komut satırı ve API aracılığıyla VIP (Virtual IP) ve Havuz ve Sunucu nesnelerinin gözlemlenmesi, vSphere Web Client vasıtasıyla VIP sessionlarının global istatistiklerinin gözlemlenmesi. | Komut satırı ve API aracılığıyla VIP (Virtual IP) ve Havuz ve Sunucu nesnelerinin gözlemlenmesi, vSphere Web Client vasıtasıyla VIP sessionlarının global istatistiklerinin gözlemlenmesi. |
| Layer 7 Manipulasyonu | Yok | URL Block, URL rewrite, content rewrite |

Tablo 74 - Yük Dengeleyici Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-SDN-032 | NSX Yük Dengeleyici kullanılacaktır. | Yönetim uygulamaları için gereksinimi NSX yük dengeleyici karşılayabilir. Başka bir yük dengeleyici kullanılması maliyetin artmasına ve SDDC tarafında yönetilmesi gereken bir bileşen daha olması demektir. | Yoktur. |
| SDDC-SA-SDN-033 | Tüm yönetim uygulamaları için NSX yük dengeleyici HA kipinde kullanılacaktır. | Yük dengeleyici gerektiren tüm yönetim uygulamaları tek bir sanal bağa sahiptir. Tek bir yük dengeleyici kullanılması tasarımı kolaylaştırır. | Bir yönetim uygulamasının yöneticisi yük dengeleyici üzerinde değişiklik gerçekleştirebilir ve bu da başka bir uygulama üzerinde etkisi olabilir. |
| SDDC-SA-SDN-034 | Platform hizmetleri kontrolcüsü için NSX yük dengeleyicisi HA kipinde kullanılacaktır. | Yük dengeleyici kullanımı, PSC kullanan tüm uygulamaların erişilebilirliğinin artması demektir. | Fazladan idari yük getirir. |

* + - 1. Bilgi Güvenliği ve Erişim Kontrolü

Sanal ağ yönetimi için NSX Manager’ın yetki kontrolü ve kimlik doğrulama işlemleri için bir servis hesabı kullanılmalıdır.

Tablo 75 - Yetkilendirme ve Kimlik Doğrulama Yönetimi Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-SDN-035 | Uygulamadan uygulamaya olan iletişim için svc-nsxmanager adlı bir servis hesabı vCenter Server üzerinde NSX Manager ile yapılandırılacaktır. | NSX Manager sanal ağ nesnelerinin hayat döngüsünü gerçekleştirebilmek için minimum izin setine ihtiyaç duyar. İfşa olmuş hesaplar durumunda hedef uygulamalara erişimler sınırlanır. | Erişilebilirliğin garantilenebilmesi için servis hesaplarının yaşam döngüsünün idame ettirilmesi gerekir. |
| SDDC-SA-SDN-036 | vCenter Server’da svc-nsxmanager servis hesabı oluşturulurken global izinler kullanılacaktır. | Tüm vCenter Server’larda servis hesaplarının oluşturulmasını basitleştirir ve standartlaştırır. Tutarlı bir yetkilendirme katmanı oluşturulur. | Tüm vCenter Server oluşumları aynı vSphere etki alanında olmalıdır. |

* + - 1. Fiziksel İş Yüklerinin Köprülenmesi

NSX tamamıyla veri yolu ESXi hipervizörü içinde kalmak üzere VXLAN’dan Layer 2 VLAN’a köprüleme özelliklerine sahiptir. Bu köprü DLR kontrol sanal makinesinin üstünde çalıştığı ESXi sanallaştırma sunucunda çalışır. DLR başına birden çok köprü olması da desteklenmektedir.

Tablo 76 - Sanaldan Fiziksele Ara Yüz Tipi Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-SDN-037 | Tüm yönetim ve misafir sanal makineleri aksi gerekip VLAN destekli port gruplara bağlanmadıkça VXLAN sanal anahtarlara bağlanacaktır. VLAN destekli port grupları kullanılırsa VXLAN, VLAN arası yönlendirmeler yapılandırmalıdır.  Eğer ağlar arası Layer 2 komşuluk bir gereksinim ise NSX L2 köprüleme kullanarak VXLAN mantıksal anahtarlar VLAN destekli port gruplarına bağlanacaktır. | NSX L2 köprüleme sadece sanal makinelerin VLAN destekli iş yükleri ile aynı bölümde olması zorunluluğu varsa ve yönlendirme yapılamıyorsa kullanılmalıdır. Örnek: fiziksel kaynaklar veya atanmış yedekleme ağı gibi.  Hem L2 köprüleme hem de dağıtık mantıksal yönlendirme aynı VXLAN mantıksal anahtar üzerinde desteklenmektedir. | VXLAN mantıksal anahtar üzerinde bulunan sanal makinelerin trafikleri genelde yönlendirilir. Yönlendirme gerektiğinde veri yolu aktif DLR’nin üstünde çalıştığı ESXi içinde geçer. Aynı şekilde köprüleme trafiği de ESXi sanallaştırma sunucudan hipervizör seviyesinde gerçekleşir. Ölçeklendirmek gerektiğinde DLR başına birden çok köprü ya da yeni DLR oluşumları konumlandırılabilir. |

* + - 1. Bölgesel Bağlantılar

Bölgeler birbirlerine bağlı olmalıdırlar. Bağlantı tipleri noktadan noktaya bağlantı, MPLS, VPN tüneli vb. olabilir. Bu bağlantı tipi uygulamadan uygulamaya değişiklik gösterir.

Bölgeler arası bağlantı tasarımı Jumbo Frame’leri desteklemeli ve gecikme süreleri 150 ms den daha az olmalıdır.

Tablo 77 - Tesisler Arası Bağlantı Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-SDN-038 | Her pod arasında yönlendirme yapabilecek bölgeler arasında bağlantı olacaktır. | NSX, evrensel nesnelerin etkinleştirilebilmesi için vCenter’lar arası şekilde yapılandırıldığında NSX Manager’lar, ESXi sunucu VTEP’leri ve NSX kontrolcüleri ile ESXi yönetim ara yüzleri arasında bağlantı gerekir. Bölgeler arası kimlik doğrulamak için vCenter Server ve PSC tasarımı teki bir vCenter Single Sign On etki alanı gerektirir. Yönetim ve işlem iş yüklerinin taşınılırlığı bölgeler arası bağlantı gerektirir. | Bölgeler arası Jumbo frames gereklidir. |
| SDDC-SA-SDN-039 | Bölgeler arası gecikme süresi 150 ms den daha az olmalıdır. | İki vCenter Server arası vMotion ve NSX tasarımı 150ms den daha az gecikme süresi gerektirir. | Yoktur. |

* + - 1. Uygulama Sanal Ağı

VMware vRealize Automation, vRealize Operations Manager veya vRealize Orchestrator gibi yönetim yazılımları geleneksel üç katmanlı mimariden (sunum/kullanıcı ara yüzü katmanı, işlem mantığı katmanı ve veri katmanı) faydalanırlar. Bu tasarım son kullanıcı hizmetlerini sunan hizmetler için yük dengeleyicisine ihtiyaç duyar.

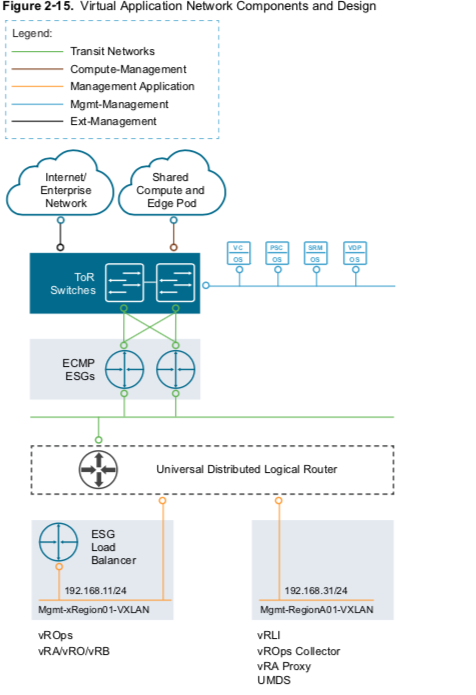
Tablo 78 - İzole Yönetim Uygulamaları Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-SDN-040 | Aşağıdaki yönetim uygulamaları uygulama sanal ağına konulacaktır.  vRealize Automation  vRealize Automation Proxy Agents  vRealize Business  vRealize Business Collectors  vRealize Orchestrator  vRealize Orchestrator Manager  vRealize Operations Manager  vRealize Operations Manager Remote collectors  vRealize Log Insight  Update Manager Download Service | Yönetim uygulamalarının erişimi sadece yayınlanan erişim noktalarından olmalıdır. | Uygulama sanal ağında NSX kenar cihazları yük dengeleme için ve dağıtık güvenlik duvarı da uygulamaları birbirlerinden ve kullanıcılardan izole etmek için kullanılır. Uygulama sanal ağına direkt erişim dağıtık güvenlik duvarı tarafından kontrol edilir. |
| SDDC-SA-SDN-041 | Üç adet uygulama sanal ağı oluşturulacaktır.  Her bölge için yönetim uygulamaları için o bölge için, yük devri gerektirmeyen adanmış uygulama sanal ağı olacaktır.  Bölgeler arası yönetim uygulama yük devri için bir uygulama sanal ağı oluşturulacaktır. | Sadece üç uygulama sanal ağı kullanmak, Layer 2 ağları ihtiyaçlarına göre uygulamalar ile paylaşmak tasarımı kolaylaştırır. | Her bir uygulama sanal ağı için tek bir /24 subnet kullanılır. Gelecekte IP adres uzayında sıkıntı yaşamamak için IP yönetimi kritik hale gelir. |

Tablo 79 - Taşınabilir Yönetim Uygulamaları Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-SDN-042 | Aşağıdaki yönetim uygulamaları bölgeler arasında kolaylıkla taşınabilir olacaktır.  vRealize Automation  vRealize Orchestrator  vRealize Business  vRealize Operations Manager | Yönetim uygulamaları yapılandırma gerektirmeden bölgeler arasında taşınabilir. | Tüm yönetim uygulamaları için benzersiz bir adresleme gerekmektedir. |

Yönetim yığınında NSX’e dayalı yazılım tanımlı ağa sahip olmak, tüm NSX özelliklerinin yönetim uygulamalarının erişebilmesine izin verir. Ağ sanallaştırma hizmetleri tasarımına bu yaklaşım güvenliği artırır, yönetim uygulamalarının taşınabilirliğini sağlar ve mevcut ağlar ile entegrasyon çabasını azaltır.

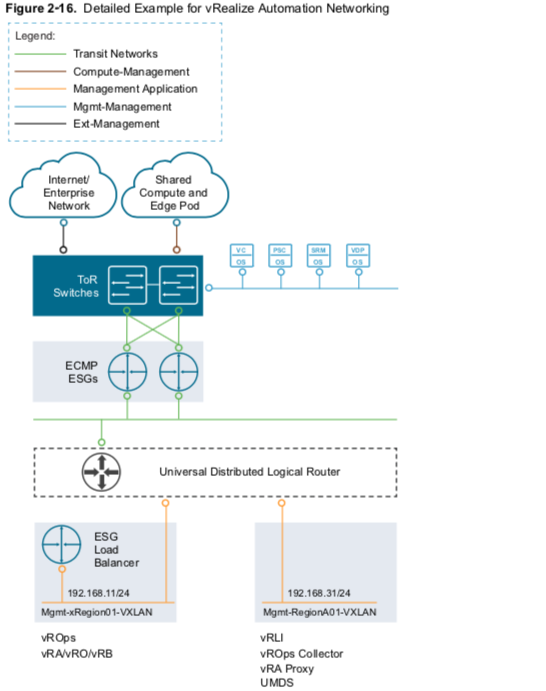


Şekil 10 - Sanal Uygulama Ağ Bileşenleri ve Tasarımı

Belirli yapılandırma seçenekleri, sonradan misafir yerleştirme süresince rahatlık sağlayabilir.

* Misafir PLR’si olarak davranacak ana bir NSX ESG ve UDLR’ye bağlantıda kullanılacak transit ağı oluşturacak bir mantıksal anahtar oluşturulacaktır.
* Ana NSX ESG uplinkleri harici ağlara bağlanacaktır.
* Ana NSX ESG dahili ara yüzleri transit ağa bağlanacaktır.
* Misafir dahili ağlarının yönlendirme kabiliyetlerini sağlamak için NSX UDLR oluşturulacak ve UDLR uplinkleri transit ağa bağlanacaktır.
* Misafir ağları UDLR’ye bağlanacaktır.
  + - 1. Sanal Ağ Tasarım Örneği

Sanal ağ tasarım örneği, bir yönetim uygulama sanal ağı uygulamasını göstermektedir. Şekil 11 de verilen örnek hizmet vRealize Automation olmakla birlikte herhangi bir üç katmanlı uygulama da benzer şekilde yapılandırılacaktır.



Şekil 11 - Detaylı vRealize Automation Ağ Örneği

Örnek kurulumu aşağıdaki şekildedir:

* vRealize Automation, bölgeler arasında yük devredebilen uygulama sanal ağı üzerinde konumlandırılır. Bu ağ VXLAN (turuncu) bağlantısı ile sağlanır.
* vRealize Automation’ın bağlı olduğu ağ harici ağlara NSX aracılığıyla gerçekleşir. NSX ESG’ler ve UDLR uygulama sanal ağları ortak ağ arasında yönlendirme yaparlar.
* vRealize Automation son kullanıcılarının erişimlerine açık olması gereken Web GUI gibi hizmetler NSX yük dengeleyicisi tarafından sağlanırlar.

Aşağıdaki tabloda uygulama sanal ağlarının IPv4 subnetlerine ilişikilendirilmeleri verilmiştir:

|  |  |  |
| --- | --- | --- |
| Uygulama Sanal Ağı | Yönetim Uygulamaları | Dahili IPv4 Subneti |
| Mgmt-xBölge01-VXLAN | vRealize Automation (Orchestrator ve Business de dahil)  vRealize Operations Manager | 192.168.11.0/24 |
| Mgmt-BölgeA01-VXLAN | vRealize Log Insight  vRealize Operations Manager Remote Collectors  vRealize Automation Proxy Agents | 192.168.31.0/24 |
| Mgmt\_BölgeB01-VXLAN | vRealize Log Insight  vRealize Operations Manager Remote Collectors  vRealize Automation Proxy Agents | 192.168.32.0/24 |

* + - 1. Güvenli Soket Katmanı (SSL) Sertifika Kullanımı

Varsayılan ayarlar ile NSX Manager kendi tarafından imzalanmış sertifikaları kullanır. Bu sertifikalar son kullanıcı cihazları ya da ağ tarayıcıları tarafından güvenilmezler. Bu sertifikaların, üçüncü partiler ya da kurumsal sertifika otoriteleri (CA) tarafından imzalanmış sertifikalar ile değiştirilmeleri tavsiye edilen ideal çalışma koşullarındandır.

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-SDN-043 | NXS Manager sertifikaları üçüncü parti Public Key alt yapısı tarafından imzalanmış sertifikalar ile değiştirilecektir. | NSX yöneticileri ile NSX Manager arasındaki iletişimin güvenilir bir sertifika ile şifrelenmiş olmasını sağlar. | Sertifikaların yenilenmesi ve değiştirilmesi fazladan bir iş yükü doğurur. |

* + 1. Paylaşımlı Veri Depolama Sistemi Tasarımı

Paylaşımlı veri depolama tasarımı vSAN depolama ve NFS depolama için olan tasarım kararlarını içerir. İyi yapılmış bir paylaşımlı veri depolama tasarımı SDDC’nin temellerini oluşturur ve aşağıdaki faydalara sahiptir:

* Kurum verilerine izinsiz erişimlerin engellenmesi
* Verilerin donanım ve yazılım arızalarına karşı korunması
* Verilerin art niyetli ya da kazayla bozulmasının engellenmesi

Ortam için paylaşımlı veri depolaması tasarlarken aşağıdaki yönlendirici ilkeler göz önünde bulundurulmalıdır:

* Uygulamaların, hizmetlerin, yöneticilerin ve kullanıcıların muhtelif ihtiyaçların karşılanabilmesi için veri depolamasının optimize edilmesi
* Maliyetlerin azaltılması, performansın artırılması, erişilebilirliğin iyileştirilmesi, güvenliğin sağlanabilmesi ve fonksiyonalitenin geliştirilebilmesi için iş gereksinimleri ile veri depolama alt yapısının stratejik olarak bir araya gelmesi.
* Uygulama veri erişiminin uygulama gereksinimlerine uyabilmesi için birden çok veri depolama katmanını sağlanması.
* Her bir veri depolama katmanının farklı performans, kapasite ve erişilebilirlik karakteristikleri ile tasarlanması gerekir. Her uygulamanın pahalı, yüksek performanslı, yüksek erişilebilirlikli veri depolamasına ihtiyacı olmadığı için farklı veri depolama katmanlarının oluşturulması maliyetleri düşürür.

Geleneksel veri depolama, VMware vSphere Virtual Volumes ve VMware vSan veri depolama arasından seçim yapılabilir. Geleneksel veri depolamada Fibre Channel, NFS ve iSCSI olgun ve sanal makine ihtiyaçlarını karşılayabilen makul bir seçenektir. VMware vSAN veri depolaması ise ESXi sanallaştırma sunucularının işlem ve veri depolama kabiliyetlerini birleştiren bir yazılım tabanlı dağıtık bir depolama platformudur. Son olarak VMware vSphere Virtual Volumes da harici bir SAN veri depolama biriminin yeteneklerinin soyutlanarak sanal alt yapıya sunulmasını sağlayan bir programlanabilir uygulama ara yüzüdür (API). Bu tasarımda Virtual Volumes kullanılmamaktadır.

* + - 1. Geleneksel Veri Depolama ve vSAN Veri Depolama

İki teknolojiden birisini ya da ötekini tercih etme kararı performans ya da yeteneklere dayanabilir ve aşağıdaki hususlar da göz önüne alınabilir:

* Kurumun kendi personelinin yetkinliği ve kurulum tabanları.
* İlk satın alma ve uzun dönem sahip olma gibi maliyetler.
* Veri depolama üreticisi ile kurumun ilişkileri.

VMware vSAN veri depolaması ise ESXi sanallaştırma sunucularının işlem ve veri depolama kabiliyetlerini birleştiren bir yazılım tabanlı dağıtık bir depolama platformudur. Kullanıcı için basit bir veri depolama yönetim deneyimi sağlar. Bu çözüm yazılım tabanlı veri depolamayı müşteri için bir gerçeğe dönüştürür. Bununla beraber bir vSAN kümesi tasarlayıp, boyutlandırırken desteklenen donanım seçenekleri dikkatle ele alınmalıdır.

* + - 1. Veri Depolama Tipleri Kıyaslaması

ESXi sanallaştırma sunucuları çeşitli veri depolama tiplerini destekler. Her bir tip veri depolama farklı vSphere özelliklerini destekler.

Tablo 80 - ESXi Sanallaştırma Sunucusunun Desteklediği Ağ Paylaşımlı Veri Depolama

|  |  |  |  |
| --- | --- | --- | --- |
| Teknoloji | Protokoller | Transferler | Ara Yüz |
| Fibre Channel | FC/SCSI | Veri/LUN blok erişimi | Fibre Channel HBA |
| Fibre Channel over Ethernet | FCoE/SCSI | Veri/LUN blok erişimi | Converged Ağ Adaptörü (Donanımsal FCoE), FCoE destekli NIC (Software FCoE) |
| iSCSI | IP/SCSI | Veri/LUN blok erişimi | iSCSI HBA ya da iSCSI etkin NIC (donanım iSCSI), Ağ adaptörü (yazılım iSCSI) |
| NFS | IP/NFS | Dosya (Direkt LUN erişimi yok) | Ağ Adaptörü |
| vSAN | IP | Veri blok erişimi | Ağ Adaptörü |

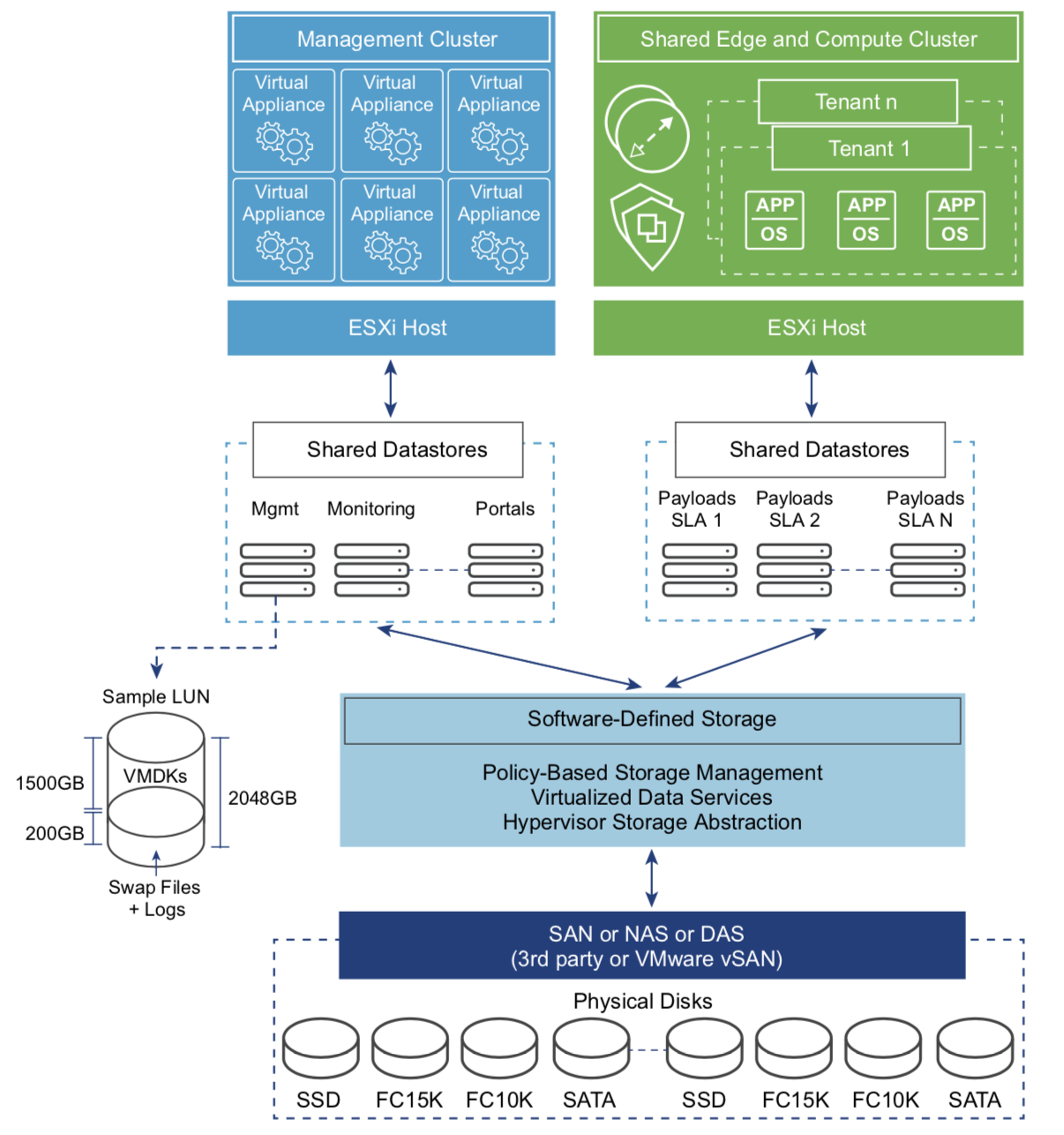
Tablo 81 - Veri Depolama Tipine göre vSphere Yetenekleri

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tip | vMotion | Datastore | Raw Device Mapping (RDM) | Uygulama ya da Blok seviyesi Kümeleme | HA/DRS | Storage API Veri Koruma |
| Yerel Depolama | Evet | VMFS | Hayır | Evet | Hayır | Evet |
| FC /FCoE | Evet | VMFS | Evet | Evet | Evet | Evet |
| iSCSI | Evet | VMFS | Evet | Evet | Evet | Evet |
| NFS | Evet | NAS | Hayır | Hayır | Evet | Evet |
| vSAN | Evet | vSAN | Hayır | Evet, iSCSI initiator ile | Evet | Evet |

* + - 1. Paylaşımlı Veri Depolama Mantıksal Tasarımı

Paylaşımlı veri depolama tasarımı her bir tip küme için uygun veri depolama cihazını belirler. Her küme tipi için kullanılacak veri depolama cihazları aşağıdaki gibidir:

* Yönetim kümeleri ana depolama cihazı olarak vSAN’ı, ikincil depolama cihazı olarak da NFS’i kullanacaktır.
* Paylaşımlı kenar ve işlem kümeleri FC/FCoE, iSCSI, NFS ya da vSAN depolamasını kullanabilir. Kullanıcı iş yükleri, SLA’ler ve diğer faktörler veri depolama ihtiyaçlarını belirlediğinden spesifik bir karar verilmemiştir.



Şekil 12 - Mantıksal Veri Depolama Tasarımı

Tablo 82 - Depolama Tipi Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-DEP-001 | Yönetim kümesinde vSAN ve NFS paylaşımlı depolama kullanılacaktır. vSAN ana depolama, NFS ise ikincil depolama olacaktır. | vSAN’ın ana depolama olarak kullanılması maliyet etkin yerel disklerin avantajını sağlar. NFS genellikle arşiv verilerin ve tarihi verilerin korunması için kullanılmaktadır. | İki ayrı veri depolama teknolojisinin kullanılması kompleksliği ve operasyonel iş yüklerini artırır. |
| SDDC-SA-DEP-002 | Tüm kümelerde, vSAN olmayan datastore’larda en az 20% boş alan olmalıdır. | Datastore’larda boş disk alanı biterse NSX kenar çekirdek ağ hizmetleri, provizyonlama portalı, VDP yedekleri ile sınırlı olmamaklar birlikte SDDC içerisindeki uygulamalar çalışmayabilir. | Kapasite yönetimi ve gözlemleme kritiktir ve pro aktif olarak gerçekleştirilmelidir. |

* + - 1. Depolama Katmanlanması (Tiering)

Bütün uygulama iş yükleri aynı depolama gereksinimlerine sahip değildirler. Depolama katmanlanması, uygulama iş yüklerine göre değişen performans, güvenilirlik ve maliyet gereksinimleri göze alınarak bu farkların gerçekleştirilmesine izin verir.

Günümüz kurumsal seviyedeki veri depolama cihazlarında birden çok disk tipi ve koruma mekanizmaları bulunmaktadır. Depolama, sunucu ve uygulama yöneticileri ise ortamda konumlandırılan her bir uygulama için ideal olan depolama yapılandırmasını seçmekte zorlanmaktadırlar. Sanallaştırma da bu seçimin yapılmasını, pek çok değişik gereksinime sahip iş yükünü bir araya getirerek zorlaştırmaktadır.

En görev kritik veriler genellikle en az miktarda olup, çevrimdışı veriler ise en yüksek miktardadır. Detaylar ise organizasyondan organizasyona göre değişiklik gösterebilir.

Bir uygulama verisinin depolama katmanına karar verirken, önce uygulamanın ya da hizmetin depolama karakteristiklerine karar vermek gerekir.

* Saniyede gerçekleşen I/O operasyonları (IOPS) gereksinimleri.
* Saniyede gerçekleşen Megabyte transfer (MBps) gereksinimleri.
* Kapasite gereksinimleri.
* Erişilebilirlik gereksinimleri.
* Gecikme gereksinimleri.

Her uygulama için bu bilgiler derlendikten sonra, uygulama verisini gereksinim karakteristiklerine uyan depolama katmanına taşıyabiliriz. Bu kararı alırken mevcut hizmet seviyesi anlaşmaları da (SLA) göz önünde bulundurmak gerekir. Uygulamanın yaşam döngüsü içerisinde verileri başka katmanlara taşımak da gerekebilir.

* + - 1. VMware Depolama için Donanımsal Hızlandırma API/CLI

Daha önceden vStorage API for Array Integration (VAAI) olarak bilinen depolama için donanımsal hızlandırma API/CLI’sı ESXi sanallaştırma sunucuları ve depolama cihazları arasında iletişimi sağlayan bir set ESXCLI komutunu desteklemektedir. Bu API/CLI’ın kullanılmasının çeşitli avantajları bulunmaktadır.

API, ESXi sanallaştırma sunucusunun belirli depolama operasyonlarını cihaza devretmesini sağlayan depolama komutlarını barındırır. Bu yük devri ESXi sunucusunda kaynaklara ek yük gelmesini engeller ve depolama ağırlıklı operasyonlarda (klonlama, sıfırlama vb.) ciddi bir performans kazancı sağlar. Donanım hızlandırmasındaki amaç, VMware I/O işlemlerine donanım desteği sağlayarak daha etkin olarak gerçekleştirmektir.

VAAI kullanılmadan sanal makinelerin klonlama veya verilerinin göç ettirilmesi VMkernel veri taşıyıcı tarafından gerçekleştirilen bir yazılımsal veri hareketi ile mümkündür. Veri taşıyıcı kaynak datastore’dan okuma ve hedef datastore’a yazma I/O komutlarını yayınlar. VAAI ile veri taşıyıcı API komutlarını kullanarak mümkün olduğunca bu operasyonları depolama cihazına devreder. Örnek vermek gerekirse, aynı veri depolama cihazı içerisinde bulunan bir datastore’dan başka bir datastore’a bir sanal makinenin disk dosyasını (VMDK) kopyalamaya çalıştığımızda veri taşıyıcı depolama cihazını yöneterek tüm kopyalama işleminin cihaz içerisinde gerçekleşmesini sağlar. Bir veri kaydırma operasyonu başlatıldığında karşılık gelen donanımsal boşaltma (offload) operasyonu etkin ise, veri taşıyıcı önce bu özelliği kullanmaya çalışacak, operasyon başarısız olursa geleneksel yöntemler ile veri kaydırmayı gerçekleştirecektir.

Hemen hemen tüm durumlarda donanımsal olarak veri taşınması, yazılımsal olarak yapılmasına kıyasla daha performanslı gerçekleşmektedir. Daha az işlemci zamanı harcanacak ve daha az bant genişliği kullanılacaktır.

Tablo 83 - Depolama Cihazları vStorage API Entegrasyonu Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-DEP-003 | NAS cihazları yerine VAAI’yi destekleyen depolama cihazları tercih edilecektir. | VAAI görevleri depolama cihazına delege ederek ESXi hipervizörün kaynaklarını uygulama iş yükleri için kullanmasını sağlar. | Tüm VAAI cihazları VAAI over NFS’i desteklemezler. Bu yeteneğin etkinleştirilebilmesi için üreticinin sağlayacağı bir eklenti gereklidir. |

* + - 1. Sanal Makine Depolama Politikaları

Sanal makineler için hangi veri depolama yeteneklerinin ve karakteristiklerinin en uygun olduğunu belirleyen politikalar belirlenebilir. vSAN depolama politikalarını kullanarak sanal makinelerin karakteristiklerinin belirlenmesine izin verir. Bu sayede politikalar vSAN birimleri üzerinde değil, sanal makine diski seviyesinde belirlenebilir. VMware vSphere API for Storage Awareness (VASA) kullanarak ya da son kullanıcı tarafından tanımlanan depolama politikaları ile depolama cihazının kabiliyetleri tanımlanabilir.

vSphere API for Storage Awareness (VASA) ile depolama cihazı üreticileri, ürünlerinin yeteneklerini vCenter Server’a yayınlayabilirler ve bu yetenekler ara yüzde sunulabilir. VMware Storage Policy SDK, PowerCLI ya da vSphere Web Client aracılığıyla son kullnıcı tanımlı depolama politikaları da tanımlanabilir.

Bir sanal makineye depolama politikaları atanabilir ve periyodik olarak uyumluluğu kontrol edilerek sanal makinenin doğru performans ve kabiliyetteki depolama üzerinde çalışmaya devam edebilir. Bir sanal makineye depolama politikası, o sanal makinenin klonu çıkartılırken ya da göç ettirilirken atanabilir. Eğer bir sanal makineye bir politika atanmışsa vSphere Web Client o politikaya uygun olan datastore’ların listesini gösterir. Eğer politikaya uygun olmayan bir datastore seçilirse uyumsuzluk rapor edilir.

Tablo 84 - Sanal Makine Depolama Politikaları Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-DEP-004 | Yönetim kümesi içerisindeki tüm sanal makineler için vSAN var sayılan depolama politikası kullanılacaktır. | Var sayılan vSAN depolama politikası yönetim sanal makineleri için yeterlidir. | Eğer üçüncü parti ya da ek sanala makinelerin farklı depolama gereksinimleri varsa ek sanal makine depolama politikaları gerekebilir. |

* + - 1. vSphere Depolama I/O Kontrol Tasarımı

Depolama I/O kontrolü küme genelinde depolama I/O önceliklendirmeden sorumludur. Bu sayede iş yükü konsolidasyonu sağlanır ve aşırı provizyonlamadan kaynaklanan ekstra maliyetleri azaltır.

I/O kontrolü hisseler ve limitler gibi yapıları depolama üzerinde değerlendirir. I/O sıkışması sırasında sanal makinelerin gerçekleştirebileceği I/O miktarlarını kontrol ederek daha önemli sanal makinelerin daha az önemdeki sanal makinelere kıyasla daha fazla I/O kaynağına erişmesini sağlar.

Bir datastore üzerinde I/O kontrolü etkinleştirildiğinde, ESXi sanallaştırma sunucusu cihaz ile iletişim gerçekleştirirken gecikme değerlerini gözlemlemeye başlar. Gecikme değerleri belirli bir eşik değerini aştığında o datastore sıkışıklık yaşıyor olarak işaretlenir ve o datastore’a erişen her bir sanal makineye hisseleri oranında I/O kaynağı tahsis edilir.

vSphere depolama I/O kontrolünün çeşitli gereksinimleri, sınırlamaları ve kısıtlamaları vardır:

* I/O kontrolünün etkinleştirildiği datastore’lar tek bir vCenter Server tarafından yönetilmelidir.
* Depolama I/O kontrolü sadece Fibre Channel, iSCSI ve NFS bağlı depolama cihazları üzerinde desteklenmektedir. RDM (Raw Disk Mapping) desteklenmemektedir.
* Depolama I/O kontrolü birden fazla uzantısı (extent) olan datastore’larda desteklenmemektedir.
* Otomatik katmanlandırma yeteneğine sahip depolama cihazları ile I/O kontrolü bir arada kullanılmadan önce VMware uyumluluk kılavuzundan kontrolü yapılmalıdır.

Tablo 85 - Depolama I/O Kontrolü Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-DEP-005 | vSAN olmayan tüm datastore’lar üzerinde I/O kontrolü var sayılan değerleri ile etkinleştirilecektir. | Depolama I/O kontrolü bir datastore üzerindeki tüm sanal makinelerin eşit miktarda I/O yapabilmesini izin verir. | Fazla I/O yapan sanal makineler diğer sanal makinelerin de I/O yapabilmesi için sadece sıkışıklık yaşandığı zamanda yavaşlatılırlar. |

* + - 1. Depolama Kümesi Tasarım Kararları

Datastore kümesi kaynaklarını ve yönetim ara yüzlerini paylaşan bir datastore koleksiyonudur. Normal kümeler ESXi sanallaştırma sunucusu için neyse datastore kümeleri de datastore’lar için odur. Bir datastore oluşturulduktan sonra vSphere DRS ile depolama kaynakları yönetilebilir.

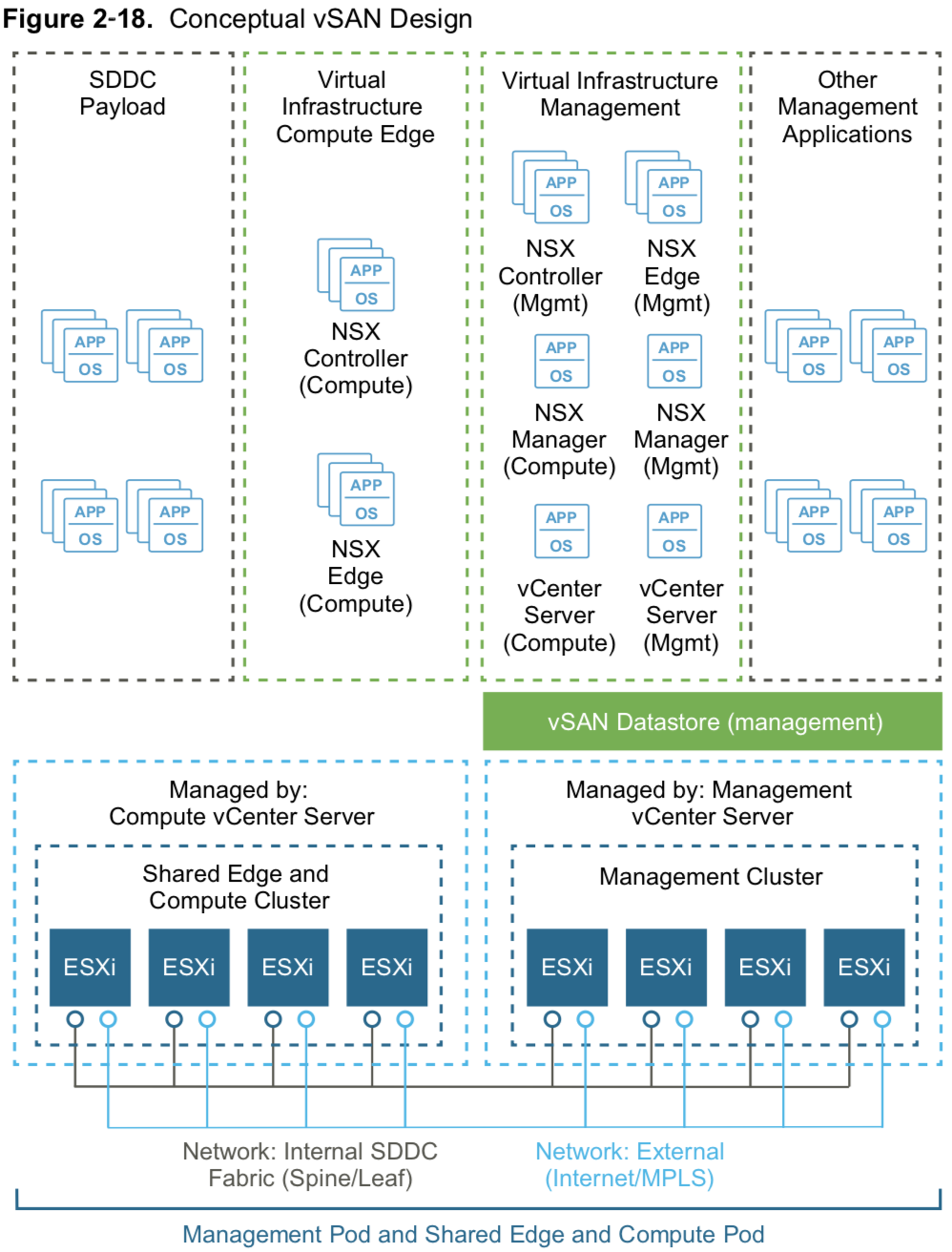
vSphere datastore kümesleri benzer datastore’ları bir depolama kaynak havuzunu içerisinde gruplarlar. Bir datastore kümesi için DRS etkinleştirildiğinde, vSphere sanal makinelerin ilk yerleştirilme sürecini otomatikleştirir ve sıkışıklıkları önlemek için küme içerisinde depolama kaynaklarını dengelemeye çalışır. Göç ettirilmeyle ilgili tavsiyeleri oluştururken, DRS datastore üzerindeki boş alanı ve I/O yükünü göz önüne alır.

Bir datastore’u datastore kümesine eklerken, o datastore’un kaynakları kümenin kaynaklarına eklenir. Aşağıdaki yönetim kabiliyetleri küme içerisindeki her bir datastore için kullanılabilir:

* Alan kullanımı yük dengelenmesi: Alan kullanımı için bir eşik değer belirlenebilir. Bu eşik değer aşıldığında DRS dengelemek amacıyla ya göç tavsiyelerinde bulunur ya da bu göç işlemlerini gerçekleştirir.
* I/O Gecikme yük dengelenmesi: Sıkışıkları önlemek için I/O gecikmeleri için bir eşik değer tanımlanabilir. Bu eşik değer aşıldığında DRS dengelemek amacıyla ya göç tavsiyelerinde bulunur ya da bu göç işlemlerini gerçekleştirir.
* Karşıt ilginlik kuralları: Sanal makinelerin diskleri için, farklı datastore’larda barınmalarını sağlamak adına karşıt ilginlik kuralları tanımlanabilir. Var sayılan ayarlar ile tüm sanal makine diskleri bir arada bulunurlar.
  + - 1. vSAN Depolama Tasarımı

Burada verilen VMware vSAN tasarımı, kavramsal tasarımı, mantıksal tasarımı, ağ tasarımını, küme ve disk grup tasarımını ve politika tasarımını içermektedir.

Bu VMware vSAN tasarımı yönetim kümesi ile sınırlıdır. Küme içerisinde yedeklilik ve performans sağlamak için var sayılan depolama politikasını kullanır. Her ne kadar vSAN paylaşımlı kenar ve işlem kümesi için kullanılabilse de bu tasarımda uygulamasına girilmemiştir.

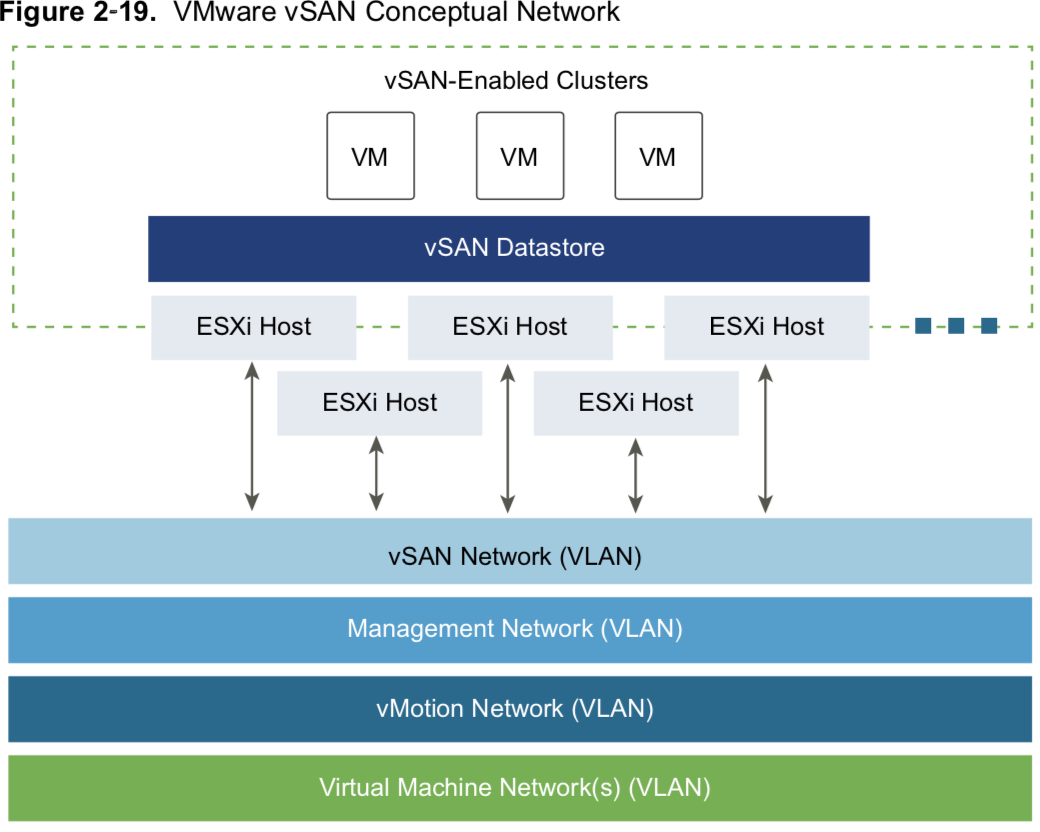


Şekil 13 - Kavramsal vSAN Tasarımı

Bir vCenter Server tarafından yönetilen küme içerisinde, işlem kaynakları nasıl yönetilebiliyorsa yazılım tanımlı depolama kaynakları da aynı şekilde yönetilebilir. İşlemci ya da hafıza rezervasyonları, limitleri ve hisseleri yerine depolama politikaları tanımlanabilir ve sanal makinelere atanabilir. Bu politikalar depolama karakteristiklerini belirler ve iş gereksinimleri değiştikçe onlar da değişebilirler.

Ağ yapılandırılması gerçekleştirilirken, trafik göz önünde bulundurulmalı ve vSAN trafiğinin nasıl izole edileceğine karar verilmelidir. Sanallaştırma sunucuları arasında ne kadar replikasyon ve iletişim trafiği olacağı düşünülmelidir. vSAN için trafik miktarı, kümede kaç adet sanal makine bulunduğu ve üzerlerindeki uygulamaların ne kadar yazma yoğunluklu I/O gerçekleştirdiklerine bağlıdır. vSAN trafiği kendi ayrı Layer 2 ağ bölümüyle izole edilmelidir. Bu iş için ayrılmış ağ anahtarları kullanılabileceği gibi portlar ya da bir VLAN ile de gerçekleştirilebilir.

vSAN VMkernel port grubu küme oluşumu esnasında yaratılır. Bu port grubunu depolama kaynaklarına katkıda bulunmayanlarda dahil olmak üzere tüm sanallaştırma sunucuları üzerinde oluşturmak gerekir.



Şekil 14 - VMware vSAN Kavramsal Ağı

Ortam için en iyi ve en öngörülebilir performans (IOPS) sağlayabilmek için vSAN ile kullanımda 10Gb Ethernet bağlantıları tavsiye edilmektedir. Bu bant genişliği olmadan cihaz performansında ciddi kayıplar söz konusudur.

Tablo 86 - Ağ Hızı Seçimi

|  |  |  |  |
| --- | --- | --- | --- |
| Tasarım Kalitesi | 1Gb | 10Gb | Yorumlar |
| Erişilebilirlik | - | - | Her iki tasarımın da erişilebilirlik üstünde etkisi yoktur. |
| Yönetilebilirlik | - | - | Her iki tasarımın da yönetilebilirlik üstünde etkisi yoktur. |
| Performans | Azalır | Artar | Daha hızlı ağlar vSAN performansını artırır. |
| Kurtarılabilirlik | Azalır | Artar | Daha hızlı ağlar yeniden oluşturma ve ortam senkronizasyon performansını artırır. |
| Güvenlik | - | - | Her iki tasarımın da güvenlik üstünde etkisi yoktur. |

Tablo 87 - Ağ bant Genişliği Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-DEP-SDS-001 | vSAN trafiği için sadece 10GbE kullanılacaktır | 10 GbE performansı idealdir. | Sanallaştırma sunucuları ve ağ bileşenleri 10GbE olmak zorundadır. |

VMware vSAN standart sanal anahtar ve dağıtık anahtarı desteklemektedir. Dağıtık anahtar kullanmanın faydası gerektiğinde ağ trafiğinde öncelik sağlayabilen ağ I/O kontrolü yapılabilmesidir. Bu tasarımda bant genişliğinin garantilenebilmesi için vSAN port gruplarında vSphere Distributed Switch kullanılmaktadır.

Tablo 88 - Sanal Anahtar Tipleri

|  |  |  |  |
| --- | --- | --- | --- |
| Tasarım Kalitesi | Standart Anahtar | Dağıtık Anahtar | Yorumlar |
| Erişilebilirlik | - | - | Her iki tasarımın da erişilebilirlik üstünde etkisi yoktur. |
| Yönetilebilirlik | Azalır | Artar | Dağıtık anahtar tüm sanallaştırma sunucular için merkezi olarak yönetilir. Standart anahtar her sunucuda ayrı ayrı yönetilir. |
| Performans | Azalır | Artar | Ağ I/O kontrolü gibi vSAN trafiği için performansı garanti eden yetenekler sadece dağıtık anahtarda vardır. |
| Kurtarılabilirlik | Azalır | Artar | Dağıtık anahtar yapılandırması yedeklenip geri döndürülebilir. |
| Güvenlik | Azalır | Artar | Dağıtık anahtar üzerinde güvenlik kontrolleri bulunmaktadır. |

Tablo 89 - Sanal Anahtar Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-DEP-SDS-002 | Yönetim kümesinde vSphere Distributed Switch kullanılacaktır. | Sıkışıklık durumunda vSAN trafiğine ihtiyacı olan bant genişliği garantilecektir. | Tüm trafik yolları ortak uplinkler üzerinden paylaşılmaktadır. |

VMware vSAN jumbo frame’leri desteklemektedir. Bir vSAN tasarımı eğer fiziksel ortam jumbo frame’leri desteklemekte ise ve bu yapılandırma büyük bir iş yükü oluşturmayacaksa kullanılmalıdır.

Tablo 90 - Jumbo Frames Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-DEP-SDS-003 | vSAN trafiğine adanmış VLAN üzerinde Jumbo frames kullanılacaktır. | Jumbo frames zaten vMotion ve NFS depolama trafik verimliliğini artırmak için kullanılmaktadır. | Ağ üzerindeki her bir cihaz Jumbo frames desteklemelidir. |

VMware, vSAN trafiğini kendi VLAN’ı içerisinde izole edilmesini tavsiye etmektedir. Birden fazla vSAN kümesi tasarlanırken her küme trafiği için ayrı ve kendine atanmış VLAN’lar ya da bölümler kullanmalıdır. Bu yaklaşım kümelerin birbirlerini etkilemelerini engeller ve sorun gidermekte yardımcı olur.

Tablo 91 - VLAN Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-DEP-SDS-004 | Her vSAN etkin küme için ayrı bir VLAN kullanılacaktır. | VLAN’lar trafik izolasyonu sağlarlar. | VLAN’lar sadece bir pod boyunca yayılabilirler. Her pod için yeterli sayıda VLAN tanımlanmalıdır. |

VMware vSAN, kümeler arası iletişimde kullanılan Layer 2 fiziksel ağ bölümünde etkinleştirilmiş IP çoklu yayım gerektirir. vSAN ağındaki tüm VMkernel portları IGMP kullanan bir çoklu yayım grubuna üyedirler.

vSAN kümesi oluşturulma zamanında atanan bir çoklu yayım adresine sahiptir. IGMP (v3) snooping Layer 2 coklu yayım trafiğinin belirli port gruplarına sınırlandırılmaları için kullanılır. Fiziksel ağ tasarımına göre IGMP snooping, fiziksel anahtar portlarının sadece vSAN VMkernel portları ile sınırlı kalması için sorgulayıcı limiti ile yapılandırılır. Bazı durumlarda IGMP snooping sorgulayıcısı belirli bir VLAN ile ilişkilidir.

Küme ve disk grubu tasarımı düşünüldüğünde, vSAN datastore büyüklüğü küme içerisindeki sanallaştırma sunucu sayısı, sunucu başına disk grubu sayısı ve vSAN politikası konularında kararlar verilmelidir.

VMware vSAN datastore büyüklüğü, datastore gereksinimine bağlıdır. Boyutlandırırken maliyet ve erişilebilirlik arasında bir denge olmalıdır.

Tablo 92 - VMware vSAN Datastore Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-DEP-SDS-005 | Yönetim kümesi için en az vSAN’da 8TB ham kapasite sağlanacaktır. | vSAN kullanan yönetim kümesi sanal makineleri en az 8 TB ham alana ihtiyaç duyarlar. NFS sadece yedekleme ve log arşivleri için kullanılmaktadır. | Yoktur |
| SDDC-SA-DEP-SDS-006 | Her zaman tüm vSAN datastore’larında en az %30 boş alan bulunacaktır. | vSAN %80 doluluğa ulaştığında yeniden dengeleme işine başlayacaktır ve buda çok kaynak kullanan bir işlemdir. | İhtiyaç duyulan depolama alanını artırır. |

Küme içerisinde bulunan sanallaştırma sunucularının sayısı, vSAN datastore’unda bulunan boş alan miktarına ve küme içerisinde tahammül edilebilen sunucu arızası sayısına bağlıdır. Örnek vermek gerekirse, eğer kümede sadece 3 ESXi sanallaştırma sunucusu varsa sadece bir arızaya tahammül edilebilir. Daha fazla erişilebilirlik isteniyorsa ek sanallaştırma sunucuları gereklidir.

Tablo 93 - Küme başına Sanallaştırma Sunucu Sayısı

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tasarım Kalitesi | 3 Sunucu | 32 Sunucu | 64 Sunucu | Yorumlar |
| Erişilebilirlik | Azalır | Artar | Artar | Kümede ne kadar çok sunucu varsa o kadar çok arızaya tahammül edilebilir |
| Yönetilebilirlik | Azalır | Artar | Artar | Kümede ne kadar çok sunucu varsa o kadar çok sanal makine konumlandırılabilir. |
| Performans | Artar | Azalır | Azalır | Eğer kaynaklarda bir dengesizlik varsa büyük bir küme performansı kötü yönde etkileyebilir. |
| Kurtarılabilirlik | - | - | - | Tasarımların kurtarılabilirlik üstünde etkisi yoktur. |
| Güvenlik | - | - | - | Tasarımların güvenlik üstünde etkisi yoktur. |

Tablo 94 - Küme Boyu Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-DEP-SDS-007 | Yönetim kümesi vSAN destekleyen en az 4 sanallaştırma sunucusundan oluşacaktır. | 4 sanallaştırma sunucusunun olması hem erişilebilirlik hem de boyut gereksinimlerini karşılamaktadır. vSAN küme sağlığı etkilenmeden bir ESXi sunucusunun bakım kipine alınıp güncelleme yapılması da sağlanır. | Yönetim kümesinin erişilebilirlik gereksinimleri küme üyelerinin yeteri kadar kullanılmamasına yol açabilir. |

Bölüm tasarımında disk grubu büyüklüğü önemli bir faktördür. Küme içerisinde ne kadar fazla sanallaştırma sunucusu varsa kümede o kadar çok arızaya tahammülü vardır. Bu yetenek disk grupları için fazladan donanım gerektirdiğinden maliyeti artırır. Daha çok disk grubu olması arıza durumunda vSAN’ın kurtarılabilirliğini artırır. Disk gruplarının sayısı belirlenirken vSAN datastore’undaki boş alan miktarı ve kümedeki tahammül edilebilen arıza sayısı göz önünde bulundurulmalıdır.

Disk gruplarının optimal sayısı, vSAN datastore’u için olan donanım ve boş alan arasında bir denge sağlanarak belirlenir. Daha fazla disk grubu olması müsait disk alanı miktarını artırır ve daha yüksek erişilebilirlik sağlar. Bununla beraber disk grupları eklemek maliyeti de yükseltir.

Tablo 95 - Sanallaştırma Sunucu başına Disk Grupları

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tasarım Kalitesi | 1 Disk Grubu | 3 Disk Grubu | 5 Disk Grubu | Yorumlar |
| Erişilebilirlik | Azalır | Artar | Artar | Kümede ne kadar çok sanallaştırma sunucusu varsa o kadar çok arızaya tahammül edilebilir. Bu yetenek ek donanımlar gerektirdiğinden maliyeti artırır. |
| Yönetilebilirlik | - | - | - | Küme de ne kadar çok sanallaştırma sunucusu varsa o kadar çok sanal makine vSAN ortamında yönetilebilir |
| Performans | - | Artar | Artar | Eğer flash ile depolama kapasitesi oranı yüksekse vSAN artan bir performans ve hız sağlar |
| Kurtarılabilirlik | - | Artar | Artar | Arıza durumunda daha fazla disk gurubu bulunması kurtarılabilirliği artırır. |
| Güvenlik | - | - | - | Tasarımların güvenlik üstünde etkisi yoktur. |

Tablo 96 - Sanallaştırma Sunucusu başına Disk Grubu Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-DEP-SDS-008 | Yönetim kümesindeki sanallaştırma sunucuları için vSAN tek bir disk grubu ile yapılandırılacaktır. | Tek bir disk grubu datastore için gerekli olan performans ve kullanılabilir alanı sağlayacaktır. | Bir sanallaştırma sunucusu üzerindeki bir SSD’nin kaybı tüm disk grubunun devre dışı kalmasına yol açar. İki ya da daha fazla disk grubu kullanılması erişilebilirlik ve performansı artırır. |

VMware vSAN etkinleştirilip, yapılandırıldıktan sonra sanal makinelerin depolama karakteristiklerini tanımlayan depolama politikaları oluşturulabilir. Depolama karakteristikleri farklı sanal makineler için farklı hizmet seviyelerini tanımlar. Var sayılan depolama politikası tek bir sunucu arızasına tahammül edilmesi ve tek bir şeride sahip olunmasıdır. Ortam aksini gerektirmedikçe varsayılan politikalar kullanılmalıdır. Eğer özel bir politika yapılandırılırsa vSAN onu garanti eder. Eğer vSAN bir politikayı garanti edemezse o politika ile sanal makine provizyonlanmasına izin vermez.

Bir depolama politikası tek başına ya da bir arada kullanılarak farklı hizmet seviyeleri oluşturan çeşitli niteliklere sahiptir. Politikalar tüketilen alan ile kurtarılabilirliği dengeleme adına erişilebilirlik ve performans için yapılandırılabilir. Pek çok durumda var sayılan sistem politikaları yeterlidir ve yeni özel politikalara ihtiyaç olmaz. Politikalar uygulamaların iş gereksinimlerine göre özelleştirilebilir.

Tablo 97 - VMware vSAN Politika Seçenekleri

|  |  |  |  |
| --- | --- | --- | --- |
| Yetenek | Kullanım Durumları | Değer | Yorumlar |
| Tahammül edilen arıza sayısı | Yedeklilik | Varsayılan 1, Maksimum 3 | Sanal makineler için standart RAID 1 aynalaması. N arıza için N+1 kopya disk oluşturulur. 2N+1 sunucu gereklidir. |
| Nesne başına disk şerit sayısı | Performans | Varsayılan 1, Maximum 12 | Sanal makineler için standart RAID 0 şerit yapılandırması. |
| Flash okuma tampon rezervasyonu (%) | Performans | Varsayılan 0, Maksimum %100 | Mantıksal nesne için ayrılan flash ön tampon oranı. Mecbur kalınmadıkça kullanılmamalıdır çünkü rezerve edilen flash ön tampon diğer sanal makineler tarafından kullanılamamaktadır. |
| Nesne alan rezervasyonu (%) | Thick Provisioning | Varsayılan 0, Maksimum %100 | Depolama nesnesinin ne oranda thick provizyonlanacağı. Depolamanın geri kalan yüzdesi thin provizyonlanacaktır. |
| Provizyonlama zorlama | Geçersiz kılma politikası | Varsayılan Hayır. | Provizyonlama zorlaması mevcut küme kaynakların ilgili politikayı sağlayamadığı durumlarda provizyonlama için kullanılır. vSAN kümesinde planlanan bir genişleme olması durumunda sanal makinelerin provizyonlanmasıne devam edilmesi içindir. Kaynaklar müsait oldukça vSAN nesneleri politikaya uygun hale getirir. |

Varsayılanda politikalar uygulama gereksinimlerine dayanarak yapılandırılırlar. Ancak nesneye bağlı olarak farklı olarak uygulanırlar.

Tablo 98 - Nesne Politikaları Varsayılan Değerleri

|  |  |  |
| --- | --- | --- |
| Nesne | Politika | Yorumlar |
| Sanal Makine isim uzayı | Arızaya Tahammül:1 | Yapılandırılabilir. Değişiklik tavsiye edilmez. |
| SWAP | Arızaya Tahammül:1 | Yapılandırılabilir. Değişiklik tavsiye edilmez. |
| Sanal Disk(ler) | Kullanıcı tanımlı depolama politikası | Sistem üzerinde yapılandırılmış herhangi bir depolama politikası olabilir |
| Sanal Disk Snapshot’ları | Sanal disk politikasını kullanır | Varsayılanda disk politikası ile aynıdır. Değişiklik tavsiye edilmez. |

Politika tasarımı iş ihtiyaçları ve uygulama gereksinimlerinin analizi ile başlar. Gerekli politikalara karar verebilmek için VMware vSAN kullanım senaryolarının değerlendirilmesi gerekir. Aşağıdaki uygulama gereksinimlerin değerlendirilerek başlanması uygun olur.

* I/O performansı ve her bir sanal disk üzerindeki iş yüklerinin profilleri.
* İş yüklerinin çalışan setleri.
* Ek ön tamponun kesintisiz eklenmesi (ön tamponun yeniden doldurulması gerekir)
* Çeşitli uygulama en iyi uygulama örnekleri (Blok boyutu gibi)

Değerlendirmeden sonra yazılım tanımlı depolama modül politikaları erişilebilirlik ve performans için muhafazakâr şekilde yapılandırılır. Tüketilen disk alanı ve kurtarılabilirlik dengededir. Pek çok durumda, eğer performans veya erişilebilirlik ile ilgili belirli bir gereksinim yoksa sistem politikaları yeterlidir.

Tablo 99 - Politika Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-DEP-SDS-009 | VMware vSAN varsayılan depolama politikaları kullanılacaktır. | Yönetim kümesi içinde varsayılan vSAN depolama politikaları yeterli yedeklilik sağlamaktadır. | Üçüncü parti sanal makineler yönetim kümesinde bulunuyorsa ek politikalara ihtiyaç olabilir. Bu sanal makinelerin gereksinimleri varsayılan politikaların desteklediklerinden farklı olabilir. |
| SDDC-SA-DEP-SDS-010 | Sanal makinelerin swap dosyaları VMware vSAN üzerinde sparse nesne olarak yapılandırılacaktır. | Bu ayarın etkinleştirilmesi ile swap dosyaları vSAN datastore’da sparse nesne olarak oluşturulacaktır. Böylece sadece kullanıldıklarında disk alanı işgal edeceklerdir. | VMware vSAN çalıştıran tüm ESXi sanallaştırma sunucularında bu ayarın etkinleştirilmesi idari ek yük getirebilir. |

* + - 1. NFS Depolama Tasarımı

Bu tasarımda belirli bir NFS üreticisi ya da modeli için yönlendirme yapılmamaktadır. Bahsi geçen yapılandırma ayarları ve değerleri için tercih edilecek üreticinin kılavuzlarına ve tavsiyelerine başvurmak gereklidir.

NFS (Network File System) dosya cihazlarını bir ESXi sanallaştırma sunucusuna sunarak ağ üzerinden bağlar. NFS sunucusu ya da cihazı yerel dosya sistemlerini ESXi sanallaştırma sunucularına erişilebilir hale getirir. ESXi sunucusu, RPC tabanlı bir protokol ile NFS sunucusu ya da cihazı üzerindeki metadata ve dosyalara erişir. NFS bir VMkernel portu kullanan standart bir NIC üzerinden erişilerek uygulanır.

Tek bir oturum bağlantısı üzerine dayalı olduğu için vSphere üzerinde NFS/NAS için bir yük dengelemesi yoktur. NAS cihazına bağlanan yolları birleştirerek bant genişliği artırılabilir. Ama bu sadece ağ ile ilgili arızalarda yedeklilik sağlar, NFS/NAS cihazı üzerinde gerçekleşebilecek hata durumlarında kesinti yaşanabilir.

vSphere hem sürüm 3 hem de sürüm 4.1 NFS ile uyumlu olmakla beraber tüm özellikler NFS v4.1 kullanan cihazlarda etkinleştirilemeyebilir.

Tablo 100 - NFS Sürümleri Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-DEP-NFS-001 | Tüm NFS datastore’ları için NFS v3 kullanılacaktır. | NFS v4.1 datastore’ları Depolama I/O kontrolü ve Site Recovery Manager ile desteklenmemektedir. | NFS v3 Kerberos yetkilendirmesini desteklememektedir. |

Yerel ağ üzerinde NFS trafiği şifresiz olarak yol almaktadır. Bu sebeple NFS depolama sadece güvenli ağlar üzerinde kullanılmalı ve trafik bu işe özel VLAN’lar ile izole edilmelidir. Pek çok NFS cihazı üzerinde sadece yetkili ve tanımlı IP adreslerinin dosya sistemine erişmesine izin veren yerleşik güvenlik özellikleri bulunur. İdeal çalışma koşulları içerisinde bu özellikler kullanılarak hangi ESXi sunucusunun hangi dosya sistemine erişeceği ve okuma/yazma yetkisi olup olmadığı belirlenmelidir. Bu sayede istenmeyen sunucuların dosya sistemlerine erişimi kısıtlanmış olur.

Tüm NFS dış aktarımları (export) depolama birimi üzerinde bulunan paylaşımlı klasörlerdir. Bu dış aktarımlar ESXi sunucuları ile depolama sistemi arasındaki iletişimi kontrol ederler. Tek bir birim üzerinde her biri farklı erişim kontrolleri içeren birden çok dışa aktarım olabilir.

Tablo 101 - NFS Dış Aktarım Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-SA-DEP-NFS-002 | vRealize Log Insight arşiv yönetim bileşeni için bir dışa aktarım oluşturulacaktır. | Bu yönetim bileşenlerinin depolama gereksinimleri ana depolamadan farklıdır. | Tasarımın genişletilebilmesi için dışa aktarımlar eklenebilir. |
| SDDC-SA-DEP-NFS-003 | vSphere Data Protection dışa aktarımı SDDC-FİZ-DEP-008 uyarınca kendi ayrı biriminde olacaktır. | Yedekleme I/O yoğun operasyonlardır. Aynı birim kullanılırsa ya yedekleme ya da diğer iş yüklerinin performansı kötüleşebilir. | Atanmış dışa aktarımlar yönetim iş yüküne ek getirebilir. |
| SDDC-SA-DEP-NFS-004 | Her bir dış aktarım için erişim sadece uygulama sanal makinelerinin veya sanallaştırma sunucularının erişebileceği şekilde sınırlandırılacaktır. | Erişimin sınırlandırılması veri güvenliğini artırır. | Dış aktarımların teker teker güvenli hale getirilmesi idari iş yükü getirir. |

* 1. Bulut Yönetim Platformu Tasarımı

Bulut yönetim platformu katmanı, SDDC’nin yönetim bileşenidir. Bu katman self servis portal aracılığıyla kiracıların kendi iş yüklerini oluşturmalarını otomatik iş yükü provizyonlama ile yapabildiği yerdir.

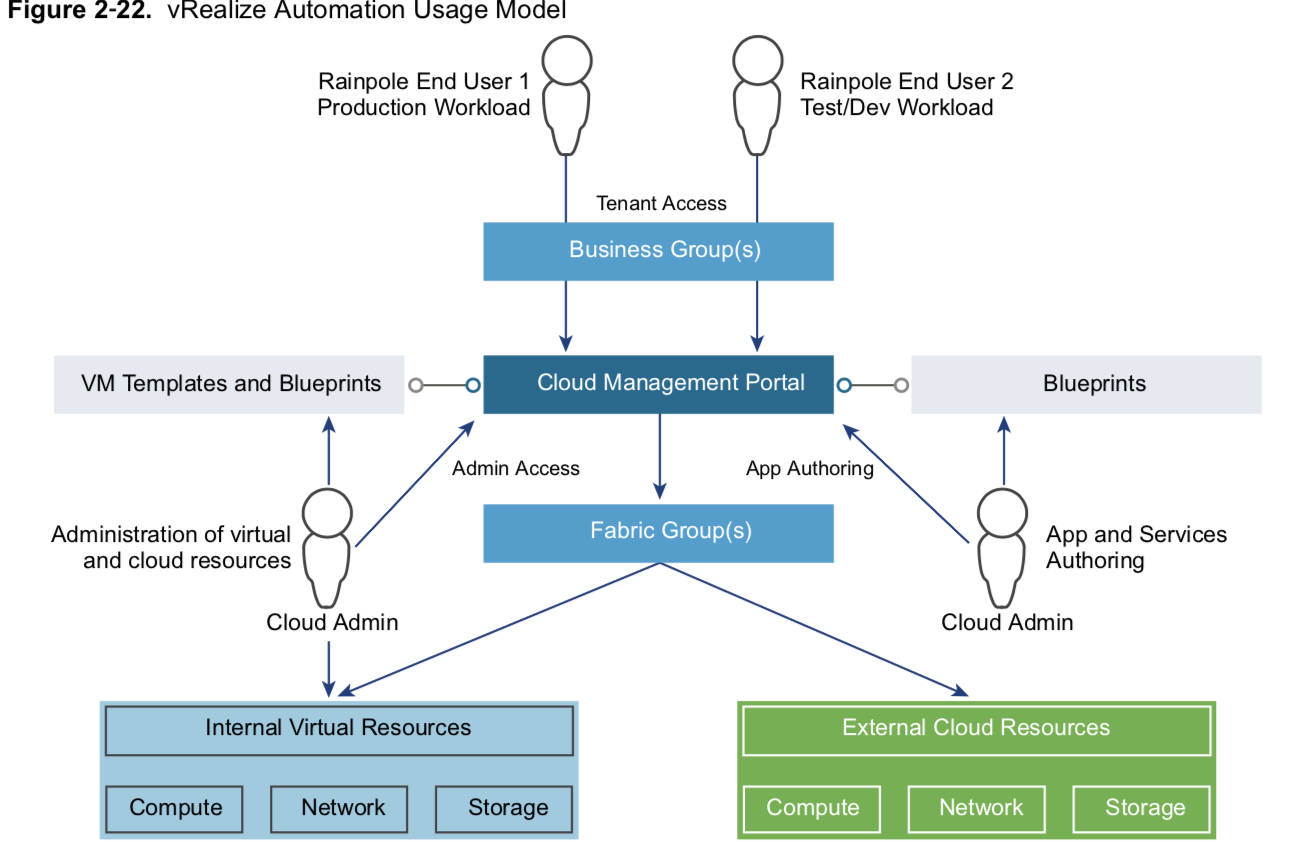
Bulut yönetim platformu (BYP) aşağıdaki bileşenlerden ve yeteneklerden oluşmaktadır:

* Hizmet Katalogu: IT hizmetlerinin tüketilmesi için tümleşik bir ara yüz sağlar. Kullanıcılar hizmet kataloğunu tarayarak IT hizmetleri ve kaynakları için istekte bulunabilir, istekleri takip edebilir ve provizyonlanmış nesneleri yönetebilirler.
* Self Servis Portalı: Self servis portalı, kullanıcıların sanal makineler ya da Amazon Web Services gibi kaynak ve IT hizmetlerini tarayıp onlar için istekte bulunabilecekleri yerdir. Bir hizmet kataloğu nesnesi için istekte bulunulduğunda o nesne ilgili bulut ortamında provizyonlanacaktır.
* Orkestrasyon: Kullanıcılar tarafından istekte bulunulan hizmet kataloğu nesneleri için otomatik iş akışları sağlar. Bu iş akışları kullanılarak VMware ve üçüncü parti teknolojiler yanı sıra SDDC alt yapısını yönetebilen süreçler otomatik olarak oluşturulup yapılandırılabilir. vRealize Automation self servis portalını ve hizmet kataloğunu sağlar. Orkestrasyon, vRealize Automation’ın içinde bulunan vRealize Orchestrator oluşumu ile etkinleşir.
  + 1. vRealize Automation Tasarımı

VMware vRealize Automation misafirlerin uygulamalar konumlandırabildiği bir hizmet kataloğu ve son kullanıcılara kişiselleştirilebilir bir portal sunar.

* + - 1. vRealize Automation Mantıksal Tasarımı

vRealize Automation pek çok kullanım senaryosunu ve entegrasyonlarını desteklemek için tasarlanmış çeşitli genişleyebilirlik seçenekleri sağlar. Buna ek olarak vRealize Automation’ın merkezi bir bileşeni olan bulut yönetim platformu kullanıcılar arasında etkileşime, bulut yönetim platformunu ile destekleyen alt yapının entregrasyonuna olana verir. Aşağıdaki şekil bulut yönetim platformu, kullanıcılar ve alt yapının birbirleri ile ilişkileri gösterilmektedir:



Şekil 15 - vRealize Automation Kullanım Modeli

Aşağıdaki tabloda ise vRealize Automation unsurları ve bu unsurları içeren bileşenler listelenmiştir:

|  |  |
| --- | --- |
| Unsur | Bileşenler |
| **Kullanıcılar** | Bulut Yöneticileri: Kurum politikaları ve organizasyonel yapı ile tanımlanmış kiracı, grup, alt yapı, hizmet ve öteki yöneticilerdir.  Bulut (ya da kiracı) Kullanıcıları: Organizasyon içindeki sanal makine provizyonlayabilen ve onlar üzerinde işletim sistemi seviyesinde direk olarak operasyon gerçekleştirebilen kullanıcılardır. |
| **Araçlar ve destekleyen alt yapı** | Sanal makine şablonları ve planları bulutun temellerini sağlayan yapı taşlarıdır. Sanal makine şablonları, son kullanıcıların bulut iş yüklerini provizyonlarken kullandıkları planları oluştururlar. |
| **Provizyonlama alt yapısı** | Dahili Sanal Kaynaklar: Desteklenen hipervizörler ve ilgili yönetim araçları  Harici Bulut Kaynakları: Desteklenen bulut sağlayıcıları ve ilgili API’lar. |
| **Bulut Yönetim Portalı** | vRealize Automation Portalı, Yönetici erişimi: Var sayılan kök kiracı portal URL’si global yapılandırma seçenekleri ve kiracı yönetimini sağlar.  vRealize Automation Portalı, Kiracı erişimi: Kiracıya özel bir tanımlayıcı ile erişilen alt kiracıyı tanımlar |

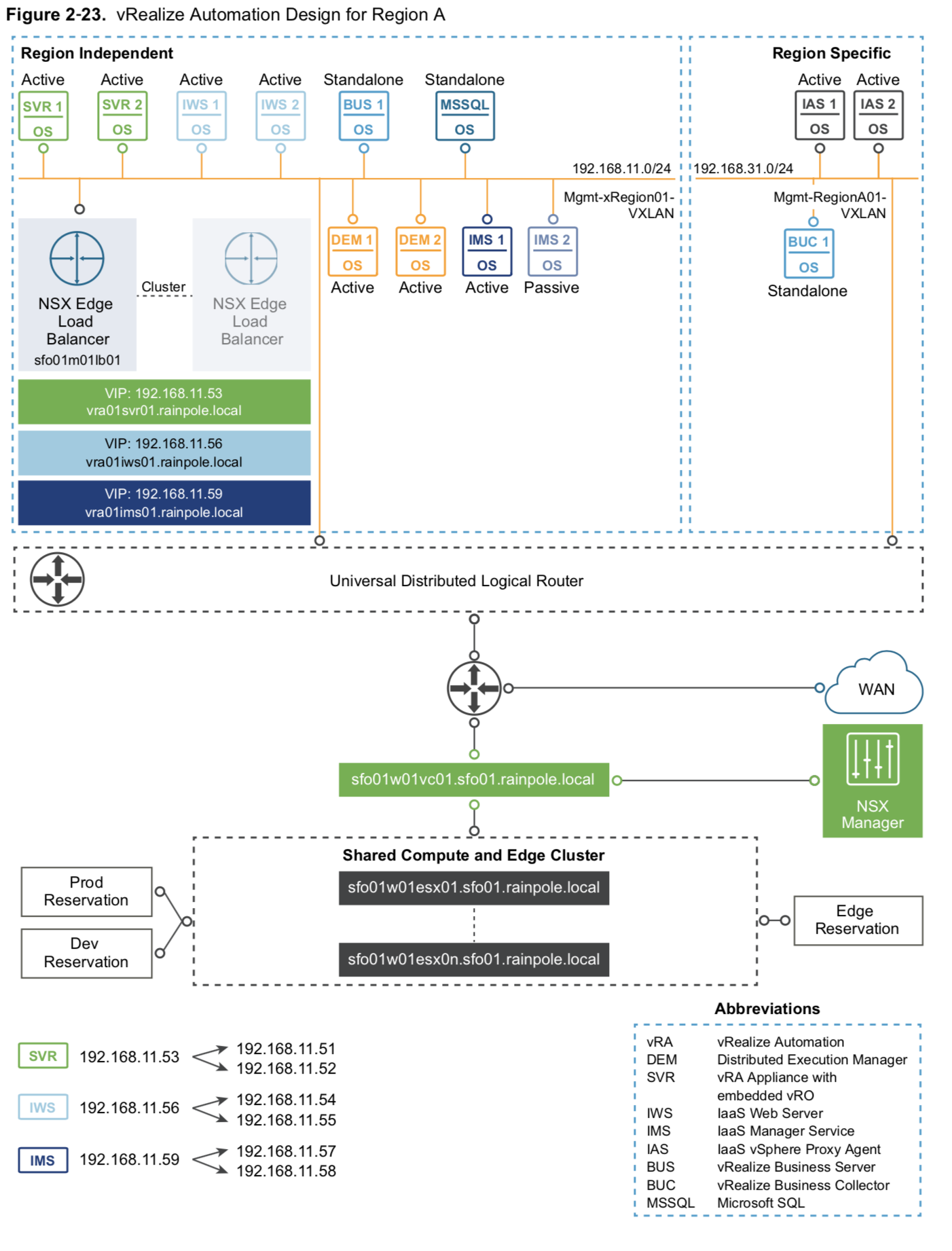
* + - 1. vRealize Automation Fiziksel Tasarımı

Fiziksel tasarım, mantıksal tasarımı destekleyen kararlardan ve karakteristiklerden oluşur. Tasarım amacı tamamıyla fonksiyonel, yüksek erişilebilirliğe sahip ve her iki bölgeye de konumlandırma yapabilen bir bulut yönetim portalı konumlandırabilmektir.

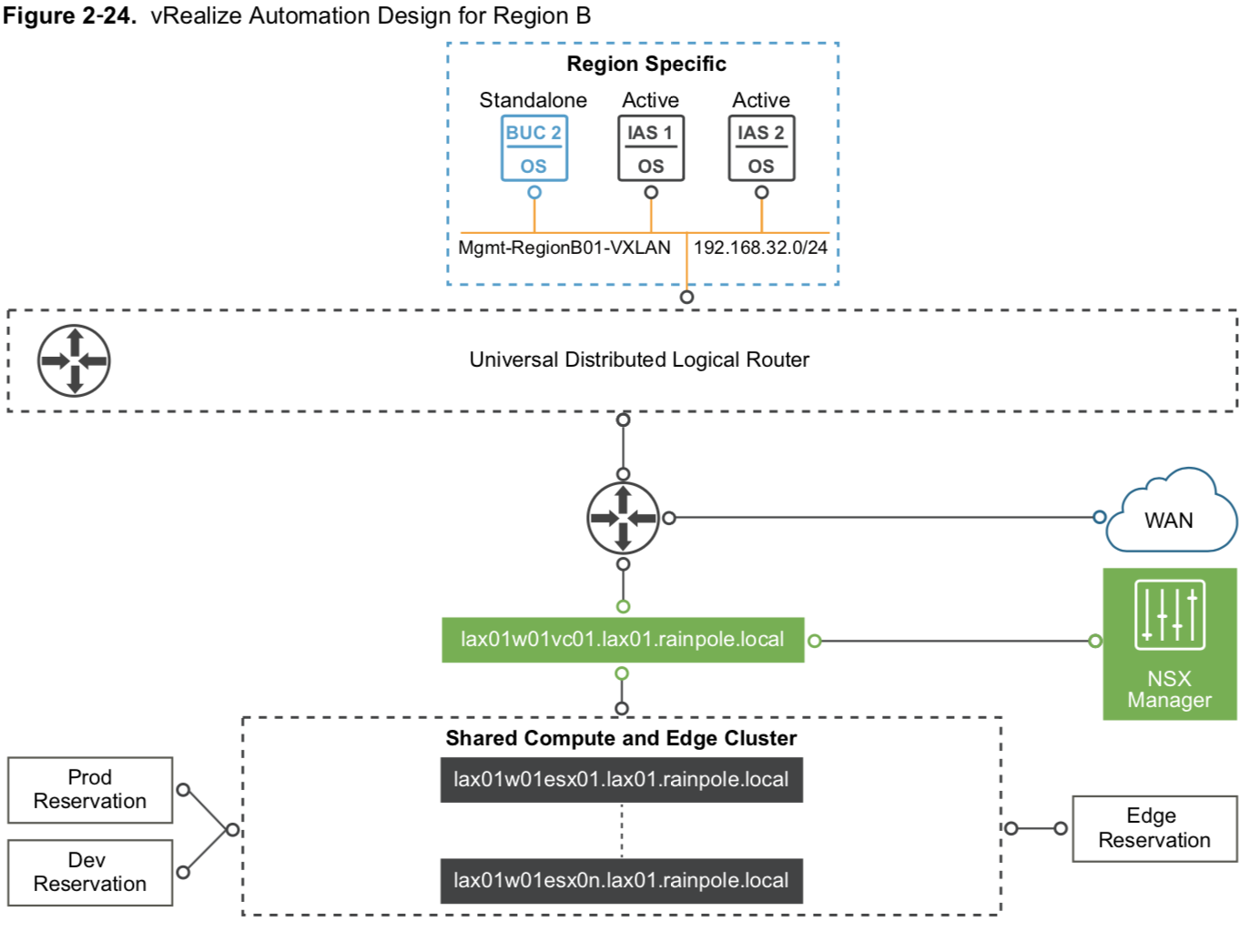
Bu tasarım hedefine ulaşabilmek için aşağıdakilerin ilk bölgeye konumlandırılmaları gerekir:

* 2 vRealize Automation Server Appliance
* 2 vRealize Automation IaaS Web Sunucuları
* 2 vRealize Automation Manager hizmet birimleri (DEM Orchestrator dahil)
* 2 DEM işçi birimi
* 2 IaaS Proxy ajan birimleri
* 1 vRealize Business for Cloud Server
* 1 vRealize Business for Cloud Remote Collector
* Microsoft SQL Server, Active Directory, DNS, NTP ve SMTP gibi destekleyen alt yapılar

vRealize Automation bileşenleri izolasyon ve yük devri amacıyla çeşitli ağ birimlerine yerleştirmek gereklidir. Tüm bulut yönetim portalı’nı oluşturan bileşenleri ağ bağlantıları ile birlikte aşağıda gösterilmektedir:



Şekil 16 - A Bölgesi için vRealize Automation Tasarımı



Şekil 17 - B Bölgesi için vRealize Automation Tasarımı

Bu tasarım vRealize Automation uygulamalarını ve destekleyen hizmetlerini soyutlayabilmek için NSX mantıksal anahtarlarını kullanmaktadır. Bu soyutlama uygulamanın ağ subnetleri, işlemci donanımları veya depolama tipleri gibi alt yapılardan bağımsız olarak konumlandırılabilmesini sağlar. Bu tasarımda vRealize Automation uygulaması ve destekleyen hizmetler A bölgesinde yer alırlar. Uygulamanın aynı oluşumu hem A bölgesindeki hem de B bölgesindeki iş yüklerini yönetebilir.

Tablo 102 - vRealize Automation Bölge Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-BYP-001 | Hem A bölgesini hem de B bölgesini yönetebilen tek bir vRealize Automation kurulumu kullanılacaktır. | vRealize Automation bir ya da daha fazla bölgeyi yönetebilir. Bu da bölgeye bakmaksızın tek bir tüketim portalı sağlar. | vRealize Automation’ın çoklu bölgeyi ele alabilecek şekilde boyutlandırılması gerekir. |

Tablo 103 - vRealize Automation Karşıt İlginlik Kuralları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-BYP-002 | vRealize Automation bileşenleri için vSphere DRS karşıt ilginlik kuralları tanımlanacaktır. | DRS kullanarak vRealize Automation birimlerinin aynı ESXi sanallaştırma sunucusunda toplanması engellenmiş olur. | Karşıt ilginlik kurallarının oluşturulması için fazladan yapılandırma gerekir. 4 ESXİ sunucudan oluşan bir yönetim kümesinde aynı anda sadece bir sunucu bakım kipine girebilir. |

Tablo 104 - vRealize Automation IaaS AD Gereksinimleri

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-BYP-003 | vRealize Automation IaaS makineleri Active Directory’e bağlanacaktır. | Bu bir zorunluluktur. | Servis hesapları ile Active Directory erişimi sağlanmalıdır. |

* + - 1. vRealize Automation Cihazı

vRealize Automation cihazı içerisinde bulut yönetimi web portalı, gömülü bir vRealize Orchestrator oluşumu ve veri tabanı hizmetleri bulunur. vRealize Automation portalı self servis provizyonlama ve bulut hizmetlerinin yönetiminin yanı sıra ayrıntılı tasarı (blueprint) yazımı, yönetimi ve denetimi sağlar. vRealize Automation sanal cihazı katalog ve veri tabanı replikasyonu için gömülü bir PostgreSQL veri tabanı kullanır. Bu veri tabanı iki vRealize Automation arasında yapılandırılarak yüksek erişilebilirlik sağlanır.

Tablo 105 - vRealize Automation Sanal Cihaz Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-BYP-004 | Yedeklilik sağlanması amacıyla iki adet vRealize Automation oluşumu konumlandırılacaktır. Her birinin içerisinde gömülü Orchestrator oluşumları olacaktır. | Yüksek erişilebilirlik için ön yüz portalı aktif/aktif olarak etkinleştirilecektir. | Yoktur. |
| SDDC-BYP-005 | Gömülü PostgreSQL veri tabanları kullanarak veri replikasyonu yapacak iki cihaz konumlandırılacaktır. | vRealize Automation için yüksek erişilebilirlik etkinleştirilecektir. Gömülü vRealize Orchestrator oluşumuları da bu veri tabanlarını kullanacaktır. | Bu aktif pasif yapılandırmada iki oluşum arasında yük devri ancak elle mümkündür. |
| SDDC-BYP-006 | Kurulum esnasında vRealize Automation cihazları 18 GB vRAM ile yapılandırılacaklardır. | 25000 adet Active Directory kullanıcısı olan ortamları destekler. | 25000’den fazla sayıda Active Directory kullanıcıları için vRealize Automation vRAM’ı 22 GB’a çıkartılmalıdır. |

Tablo 106 - vRealize Automation Sanal Cihazı Kaynak Gereksinimleri

|  |  |
| --- | --- |
| Nitelik | Değer |
| vCPU Sayısı | 4 |
| Hafıza | 18 GB |
| vRealize Automation Fonksiyonu | Portal Web Sitesi, Uygulama, Orchestrator, Hizmet Kataloğu ve Kimlik Yöneticisi |

* + - 1. vRealize Automation IaaS Web Sunucusu

vRealize Automation IaaS web sunucusu, portal içerisinden IaaS bileşenlerinin yönetimi ve tüketimi için bir kullanıcı ara yüzü sağlar. IaaS web sitesi, Automation konsoluna alt yapı yönetimi ve servis yazım kabiliyetlerini ekler. Web sitesi bileşeni DEM’den (Distributed Execution Manager), proxy ajanlarından ve veri tabanından gelen güncellemeleri sağlayan model yöneticisi ile konuşur. Model yöneticisi veri tabanı, DEM’ler ve portal web sitesi ile konuşur ve iki ayrı olarak kurulabilen bileşenden oluşur: model yöneticisi web servisi ve model yöneticisi veri bileşeni.

Tablo 107 - vRealize Automation IaaS Web Sunucusu Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-BYP-007 | İki ayrı vRealize Automation IaaS web sunucusu kurulacaktır. | vRealize Automation 1000 ila 10000 adet sanal sunucuyu desteklemektedir. İki vRealize Automation IaaS web sunucusu web bileşenlerine yedeklilik kazandırır. | Daha fazla sunucu konumlandırıldığından operasyonlara ek yük gelmiş olur. |

Tablo 108 - vRealize Automation IaaS Web Sunucusu Kaynak Gereksinimleri

|  |  |
| --- | --- |
| Nitelik | Değer |
| vCPU sayısı | 2 |
| Hafıza | 4 GB |
| vNIC port sayısı | 1 |
| Yerel disk adedi | 1 |
| vRealize Automation Fonksiyonları | Model Yöneticisi (web servisi) |
| İşletim Sistemi | Microsoft Windows Server 2012 R2 SP2 |

* + - 1. vRealize Automation IaaS Yönetim Servisi ve DEM Orchestrator Sunucusu

IaaS yönetim servisi ve DEM (distributed execution management) sunucusu, vRealize Automation IaaS platformunun çekirdeğini oluştururlar ve çeşitli fonksiyonları desteklerler:

* vRealize Automation IaaS’ın harici sistemler ve veri tabanları ile olan entegrasyonlarını yönetirler.
* DEM’lere iş mantığını aktarırlar.
* İş mantıklarını ve yürütme politikalarını barındırırlar.
* Tüm iş akışlarını ve onları destekleyen yapıları barındırırlar.

DEM özel modellerin iş mantıklarını gerektikçe diğer vRealize Automation bileşenleri ile etkileşerek çalıştırırlar. Her bir DEM oluşumu ya bir Orchestrator rolü ya da işçi roü olarak ortaya çıkarlar. DEM Orchestrator diğer DEM işçilerinin durumlarını gözlemler. Eğer bir işçi durmuş ise ya da model yöneticisi ile bağlantısını kaybetmiş ise DEM Orchestrator iş akışını tekrar kuyruğa sokar. Yeni iş akışlarını planlanan zamanlarda oluşturarak tarifeli iş yüklerini yönetir ve bir anda sadece tek bir iş akışının çalıştığından emin olur. İşletime sokmadan önce iş akışlarını ön işlemden geçirir. Ön işleme iş akışlarının gerektirdiği koşulları denetler ve iş akışının çalışma tarihçesini oluşturur.

Tablo 109 - vRealize Automation IaaS Yönetim Servisi e DEM Orchestrator Sunucusu Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-PYB-008 | Hem vRealize Automation IaaS Yöneticisi servisi hem de DEM Orchestrator’un içerisinde yük dengeleyicisi vasıtasıyla çalışacağı iki adet sanal makine oluşturulacaktır. | DEM Orchestrator’un model yöneticisi ile arasındaki bağlantıyı sağlayacak kuvvetli bir ağ bağlantısı gereklidir. | Bu iki sanal makinenin gereksinimlerini karşılamak için daha fazla kaynak gerekecektir. |

Tablo 110 - vRealize Automation IaaS Model Yöneticisi ve DEM Orchestrator Sunucu Kaynak Gereksinimleri

|  |  |
| --- | --- |
| Nitelik | Değer |
| vCPU sayısı | 2 |
| Hafıza | 4 GB |
| vNIC port sayısı | 1 |
| Yerel disk adedi | 1 |
| vRealize Automation Fonksiyonları | IaaS Yönetim Servisi, DEM Orchestrator |
| İşletim Sistemi | Microsoft Windows Server 2012 R2 SP2 |

* + - 1. vRealize Automation IaaS DEM İşçi Sanal Makinesi

vRealize Automation IaaS DEM işçileri, portal tarafından başlatılan oluşturma ya da yok etme görevlerini işletirler. DEM işçileri aynı zamanda belirli alt yapı uç noktaları ile iletişim için de kullanılırlar.

Tablo 111 - vRealize Automation IaaS DEM İşçi Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-PYB-009 | Her bir DEM sunucusu başına üç adet DEM işçisi konumlandırılacaktır. | Her bir DEM işçisi 30 adede kadar eş zamanlı iş akışı yürütebilir. Bu rakamın ötesinde iş akışları kuyrukta beklemeye alınırlar. Eğer eş zamanlı iş akışları sürekli olarak 90’nın üzerindeyse yeni DEM işçileri eklenebilir. | Fazladan DEM işçileri eklenecekse bu fazladan kaynak eklemek anlamına gelmektedir. |

Tablo 112 - vRealize Automation DEM İşçi Kaynak Gereksinimleri

|  |  |
| --- | --- |
| Nitelik | Değer |
| vCPU sayısı | 2 |
| Hafıza | 6 GB |
| vNIC port sayısı | 1 |
| Yerel disk adedi | 1 |
| vRealize Automation Fonksiyonları | DEM İşçisi |
| İşletim Sistemi | Microsoft Windows Server 2012 R2 SP2 |

* + - 1. vRealize Automation IaaS Proxy Ajanları

vRealize Automation IaaS proxy ajanları belirli alt yapı uç noktaları ile iletişimde kullanılan birer Windows servisleridir. Bu tasarımda proxy ajanları sadece vCenter sunucusu ile iletişimde kullanılmaktadır.

IaaS proxy ajanları aşağıdaki fonksiyonlara sahiptirler:

* Proxy ajanları çeşitli alt yapı bileşenleri ile iletişimde bulunurlar.
* vRealize Automation’ın kendisi kaynakları sanallaştırmaz, bunun yerine vSphere ile çalışarak sanal makineleri provizyonlar ve yönetir. vSphere’a komut göndermek ve veri almak için de proxy ajanlarını kullanır.

Tablo 113 - vRealize Automation IaaS Ajan Sunucusu Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-PYB-010 | İki adet vRealize Automation vSphere Proxy ajan sanal makinesi konumlandırılacaktır. | İki sanal makine kullanımı vSphere bağntısı için yedeklilik sağlarlar. | Birden fazla sanal makine konumlandırıldığı için daha fazla kaynak tüketilmiş olur. |
| SDDC-PYB-011 | Siteler arasında ana vRealize Automation bileşenlerinden bağımsız olarak yük devri yapılabilmesi için proxy ajan sanal makinelerinin sanal ağı ayrılarak soyutlanacaktır. | vRealize Automation oluşumlarının sitelerden bağımsız olarak yük devrine müsaade eder. | Proxy ajanları için ek uygulama sanal ağlarının ve ilişkili kenar cihazlarının konumlandırılmaları gerekir. |

Tablo 114 - vRealize Automation IaaS Proxy Ajanı Kaynak Gereksinimleri

|  |  |
| --- | --- |
| Nitelik | Değer |
| vCPU sayısı | 2 |
| Hafıza | 4 GB |
| vNIC port sayısı | 1 |
| Yerel disk adedi | 1 |
| vRealize Automation Fonksiyonları | vSphere Proxy Ajanı |
| İşletim Sistemi | Microsoft Windows Server 2012 R2 SP2 |

* + - 1. Yük Dengeleyicisi

Bir yük dengeleyicisinin oturum sürerliği, bir sunucu ile oturum açıldıktan sonra tüm isteklerin aynı sunucu tarafından cevaplanmasını sağlar. Oturum sürerliği yük dengeleyici üzerinde etkinleştirilerek her bir özgün müteakip isteklerin yük dengeleyici havuzundaki aynı vRealize Automation sunucusu tarafından cevaplanmasını sağlar. Yük dengeleyici aynı zamanda vRealize Automation sunucusunun (yönetim servisi) yük devrini de ele alır. Oturum sürerliği yönetim servisi için zorunlu olmadığı için varsayılanda etkinleştirilmemiştir.

Tablo 115 - Yük Dengeleyici Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-BYP-012 | vRealize Automation servisleri için yüksek erişilebilirlik sağlayabilmek amacıyla yük dengeleyicisi kullanılacaktır. | Daha fazla yükü kaldırabilmesi için ve daha üst erişilebilirlik için etkileştirilmesi gereklidir. | Yük dengeleyicinin etkinleştirilmesi için ek yapılandırma gereklidir. |
| SDDC-BYP-013 | Yük dengeleyicisi vRealize Automation sunucusu, uzak konsol proxy ve IaaS web sunucusu için kaynak IP tabanlı sürerliği ile round-robin algoritması ile 1800 saniye zaman aşımıyla yapılandırılacaktır. |  | Yoktur |
| SDDC-BYP-014 | Yük dengeleyicisi vRealize IaaS sunucusu ve Orchestrator için Round robin algoritması kullanarak ve sürerlilik olmadan şekilde yapılandırılacaktır. |  | Yoktur |

Aşağıda verilen vRealize Automation için yük dengeleyicisi değerleri göz önünde bulundurulmalıdır:

Tablo 116 - Yük Dengeleyici Uygulama Profili Karakteristikleri

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sunucu Rolü | Tipi | SSL düzgeçiş | Sürerlilik | Süre Bitimi (sn) |
| vRealize Automation - Sürerlilik | HTTPS (443) | Etkin | Kaynak IP | 1800 |
| vRealize Automation | HTTPS (443) | Etkin |  |  |

Tablo 117 - Yük Dengeleyici Hizmet İzleme Karakteristikleri

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| İzleme | Aralık | Zaman Aşımı | Maksimum Deneme | Tip | Beklenen | Metot | URL | Alınan |
| vRealize Automation Cihazı | 3 | 10 | 3 | HTTPS | 204 | GET | /vcac/services/api/health |  |
| vRealize Automation IaaS Web | 3 | 10 | 3 | HTTPS |  | GET | /wapi/api/status/web | REGISTERED |
| vRealize Automation IaaS Yönetim | 3 | 10 | 3 | HTTPS |  | GET | /VMPSProvision | ProvisionService |
| vRealize Orchestrator | 3 | 10 | 3 | HTTPS |  | GET | /vco-controlcenter/docs |  |

Tablo 118 - Yük Dengeleyici Havuzu Karakteristikleri

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sunucu Rolü | Algoritma | İzleme | Üyeler | Port | İzleme Portu |
| vRealize Automation Cihazı | Round Robin | vRealize Automation Cihaz monitörü | vRealize Automation cihaz birimleri | 443 | 443 |
| vRealize Automation Uzak Konsol Proxy | Round Robin | vRealize Automation Cihaz monitörü | vRealize Automation cihaz birimleri | 8444 | 443 |
| vRealize Automation IaaS Web | Round Robin | vRealize Automation IaaS Web monitörü | IaaS web birimleri | 443 | 443 |
| vRealize Automation IaaS Yöneticisi | Round Robin | vRealize Automation IaaS yönetici monitörü | IaaS yönetici birimleri | 443 | 443 |
| vRealize Automation Cihazı | Round Robin | Gömülü vRealize Automation Orchestrator kontrol merkez monitörü | vRealize Automation cihaz birimleri | 8283 | 8283 |

Tablo 119 - Sanal Sunucu Karakteristikleri

|  |  |  |  |
| --- | --- | --- | --- |
| Protokol | Port | Var Sayılan Havuz | Uygulama Profili |
| HTTPS | 443 | vRealize Automation Uygulama Havuzu | vRealize Automation – Sürerlilik profili |
| HTTPS | 443 | vRealize Automation IaaS Web Havuzu | vRealize Automation – Sürerlilik profili |
| HTTPS | 443 | vRealize Automation IaaS Yönetici Havuzu | vRealize Automation Profili |
| HTTPS | 443 | Gömülü vRealize Orchestrator Kontrol Merkezi Havuzu | vRealize Automation – Sürerlilik profili |
| HTTPS | 443 | vRealize Automation Uzak Konsol Proxy Havuzu | vRealize Automation – Sürerlilik profili |

* + - 1. vRealize Automation’da Bilgi Güvenliği ve Erişim Kontrolü

vRealize Operations yöneticisinin SDDC içerisinde sanal nesneler oluşturması ve yönetmesi için ve vRealize Automation’ın vCenter sunucusunda yetkilendirilip kimlik doğrulama yapabilmesi için bir servis hesabı kullanılmalıdır.

Tablo 120 - Yetkilendirme ve Kimlik Doğrulama Yönetimi Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-BYP-015 | vRealize Automation’ın vSphere ile uygulamadan uygulamaya konuşabilmesi için vCenter da svc-vra adlı bir servis hesabı oluşturulacaktır. | SDDC’nin bileşenleri arasındaki istek cevap etkileşimlerini takip edebilmeye yarar. | Servis hesabının yaşam döngüsü SDDC yığını dışında takip edilmelidir. |
| SDDC-BYP-016 | vCenter sunucusunda svc-vra hesap oluşturulurken yerel izinler kullanılacaktır. | Yerel yetkilerin kullanılması sadece işlem vCenter sunucu oluşumlarının geçerli ve erişilebilir vRealize Automation uç noktaları olması garantilenir. | Eğer birden fazla işlem vCenter sunucusu konumlandırılacaksa servis hesabının her bir vCenter’da yerel yetkilere sahip olduğundan emin olunmalıdır. |
| SDDC-BYP-017 | vRealize Operations Manager üzerinde svc-vra-vrops adlı bir servis hesabı oluşturularak vRealize Automation ile aralarında uygulamadan uygulamaya konuşabilmesi sağlanacak ve kiracı iş yüklerinin sağlık ve kaynak ölçümleri toplanabilecektir. | vRealize Automation, Operations Manager’a iş yüklerinin geri dönüşümü için gereken ölçümleri toplaması için en az yetki ile bağlanırlar. | Servis hesabının yaşam döngüsü SDDC yığını dışında takip edilmelidir. |

* + - 1. vRealize Automation Destek Alt Yapısı

SDDC tasarımının gereksinimlerini karşılayabilmek adına, vRealize Automation için yüksek erişilebilir veri tabanı hizmeti ve bildirimleri için elektronik posta hizmeti gibi ek bileşenlerde yapılandırılmalıdır.

vRealize Automation, IaaS elemanlarını ve yönettiği makinelerin bilgilerini saklamak için bir Microsoft SQL Server veri tabanı kullanır.

Tablo 121 - vRealize Automation SQL Veri Tabanı Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-BYP-018 | vRealize Automation’un erişilebilirlik ve I/O ihtiyaçlarını karşılayacak bir Microsoft SQL sunucu kurulacaktır. | Gereksinimleri karşıladığı sürece ister atanmış ister paylaşımlı bir SQL server kullanılabilir. | Ek kaynaklar ve lisanslar gereklidir. |
| SDDC-BYP-019 | Ya Microsoft SQL sunucuda vRealize Automation ile aynı sanal ağda olmalı ya da global bir yedek yük devredebileceği eşleniği olacaktır. | Bir bölgeden başka bir bölgeye basit bir yük devri yapılabilmesi için Microsoft SQL Server’ın da uygulama sanal ağında bir sanal makine olarak çalışması gerekir. Eğer ortam paylaşımlı bir SQL server kullanıyorsa global yük devri hem ana hem de ikincil bölgeler için erişilebilirlik sağlanır. | Microsoft SQL Server yönetimi fazladan iş yükü getirebilir. |
| SDDC-BYP-020 | Microsoft SQL Server işletim sistemi, SQL Data, Transction loğları, TempDB ve yedekler için ayrı disk alanları yapılandırılacaktır. | Veri tabanı dosyaları ve transaction loğları ve işletim sistemleri farklı disk bölümlerinde olmalıdır. | Organizasyonun veri tabanı yöneticilerine danışmak gerekebilir. |

Tablo 122 - vRealize Automation SQL Veri Tabanı Kaynak Gereksinimleri

|  |  |
| --- | --- |
| Özellik | Değer |
| vCPU Sayısı | 8 |
| Hafıza | 16 GB |
| vNIC port Sayısı | 1 |
| Yerel Disk Sayısı | 1,40 GB (D: Uygulama), 40 GB (E: Veri Tabanı Data), 20 GB (F: Veri Tabanı Log), 20 GB (G: TempDB), 80 GB (H: Backup) |
| vRealize Automation Fonksiyonları | Microsoft SQL Server Veri Tabanı |
| Microsoft SQL Sürümü | SQL Server 2012 |
| Microsoft SQL Veri Tabanu Sürümü | SQL Server 2012 (110) |
| İşletim Sistemi | Microsoft Windows Server 2012 R2 |

vRealize Automation cihazı portal elemanlarını ve hizmetlerini, cihazın yönettiği katalog öğelerinin bilgilerini tutabilmek için bir PostgreSQL veri tabanı kullanır. PostgreSQL aynı zamanda gömülü vRealize Orchestrator olulumuna dair bilgileri de içerir.

Tablo 123 - vRealize Automation PostgreSQL Veri Tabanı Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-BYP-021 | Her vRealize Automation cihazı içerisinde gömülü bulunan PostgreSQL veri tabanı kullanılacaktır. Bu veri tabanı aynı zamanda gömülü vRealize Orchetsrator tarafından da kullanılacaktır. | Tasarımı basitleştirir ve iki vRealize Automation arasında veri tabanı replikasyonunu sağlar. | Yoktur. |
| SDDC-BYP-022 | Asenkron replikasyon yapacak şekilde PostgreSQL veri tabanı yapılandırılacaktır. | Asenkron replikasyon erişilebilirlik ile performans arasında iyi bir denge sağlar. | Asenkron replikasyon tasarım hedefleri ile uyumlu bir seviyede erişilebilirlik sağlar. |

* + - 1. Bildirim Elektronik Posta Sunucusu

vRealize Automation bildirim elektronik postaları SMTP vasıtasıyla gönderilir. Bu e-postalar makine oluşturulmalarını, sürelerinin dolmalarını ve kullanıcılar tarafından alınan onay bildirimlerinden oluşurlar. vRealize Automation SMTP sunucusuna yapılan hem anonim bağlantıları hem de temel kimlik doğrulamalı bağlantıları destekler. Aynı zamanda SSL ile veya SSL’siz iletişimleri de destekler.

Onay cevapları gibi gelen e-posta bildirimlerini idare eden bir global, gelen e-posta sunucusu oluşturulur. Tüm kiracılar için varsayılan olarak görünen tek bir global gelen e-posta sunucusuna ihtiyaç vardır. E-posta sunucusu her kullanıcı için özelleştirilebilen hesapları sağlar. Her kiracı bu ayarları geçersiz kılarak kendi ayarlarını yapabilir. Eğer kiracı sistem yöneticisi bildirimleri etkinleştirmeden önce bu varsayılan ayarları geçersiz kılmazsa vRealize Automation global olarak ayarlanmış e-posta sunucusunu kullanacaktır. E-posta sunucusu POP ve IMAP protokollerini SSL’li veya SSL’siz olarak kullanabilir.

* + - 1. Bildirimler

Sistem yöneticileri sistem bildirimlerini gönderen hem gelen hem de giden e-posta sunucularının varsayılan ayarlarını yapılandırırlar. Sistem yöneticileri her bir tipten sunucudan birer tane yapılandırabilirler ve bu sunucular kiracılara varsayılan olarak görünürler. Eğer kiracılar varsayılan ayarları geçersiz kılmazlarsa bu varsayılan sunucular kullanılırlar.

Sistem yöneticileri giden e-posta bildirimleri için global bir giden e-posta sunucusu ve gelen onaylar gibi e-postaları işlemeleri için global gelen e-posta sunucusu oluştururlar.

Tablo 124 - vRealize Automation E-posta Sunucusu Yapılandırılması

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-BYP-023 | vRealize Automation için giden e-posta bildirimleri için global bir giden e-posta sunucusu ve gelen onaylar gibi e-postaları işlemeleri için global gelen e-posta sunucusu oluşturulacaktır. | vRealize Automation onayları ve sistem bildirimleri için gereklidir. | SMTP/IMAP sunucuları hazırlanmalı ve gerekli güvenlik duvarı ayarları yapılmalıdır. |

* + - 1. vRealize Automation Bulut Kiracı Tasarımı

Bir kiracı, vRealize Automation konumlandırılması bünyesinde bir organizasyonel birimdir ve bir kurum içinde bir iş birimini ya da hizmet sağlayıcısından bulut hizmetleri alan bir şirkete işaret edebilir. Her bir kiracının her ne kadar sistem seviyesindeki yapılandırmalar ortak olsa da kendisine has özgün yapılandırmaları bulunur.

vRealize Automation tek bir kiracı ya da çoklu kiracı için farklı yapılandırmaları destekler. Sistem çapındaki yapılandırmalar her zaman varsayılan kiracı kullanılarak yapılır ve bir ya da daha fazla diğer kiracılara uygulanabilir.

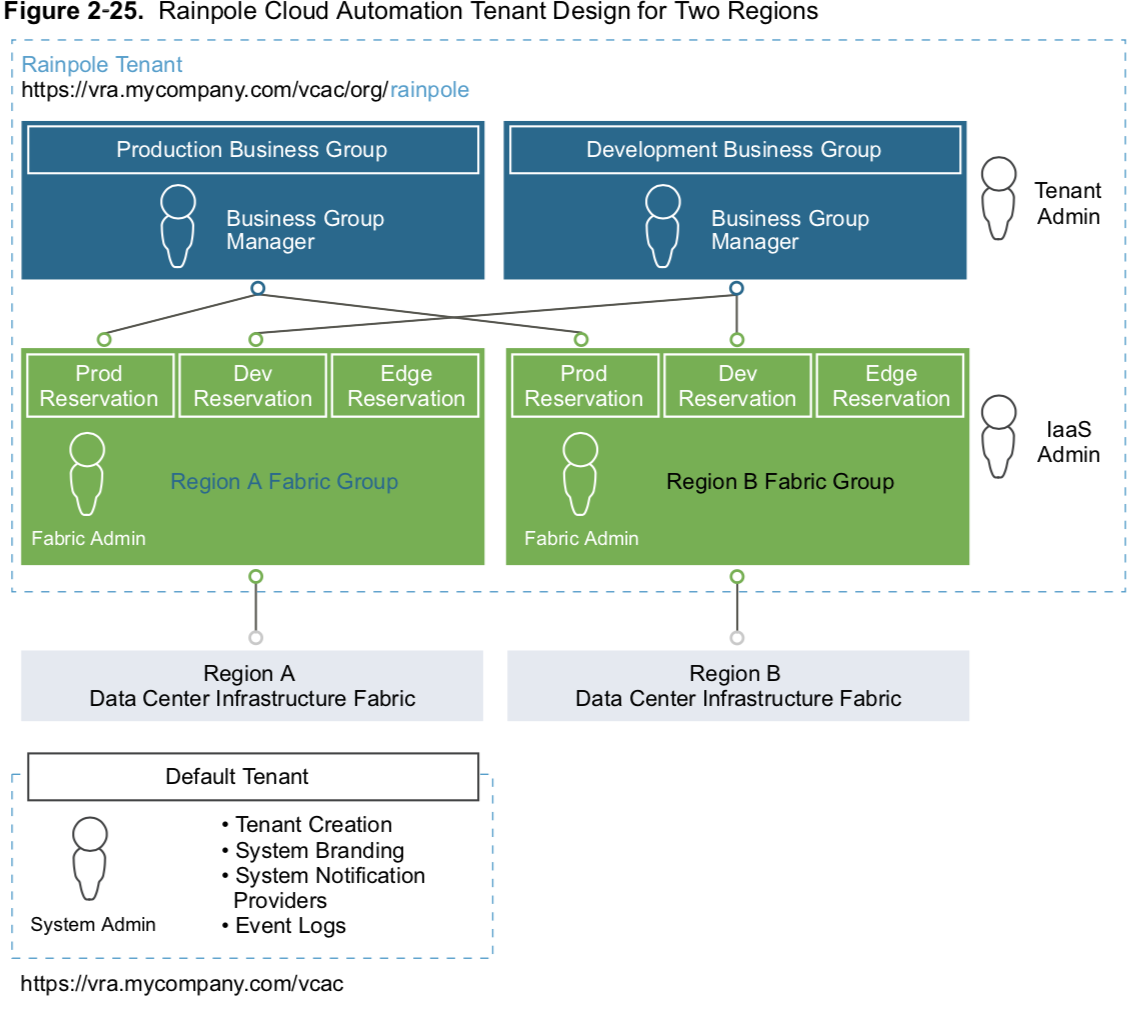
Provizyonlama için olan alt yapı kaynakları da dahil olmak üzere, alt yapı yapılandırması kiracılar için ayrı yapılandırılabilir ya da paylaşılabilir. Bulut, sanal işlem kaynakları ya da fiziksel makineler fabrik yöneticileri tarafından yönetilen fabrik gruplarına bölünebilirler. Her fabrik grubu içindeki kaynaklar rezervasyonlar kullanılmak suretiyle her bir kiracı bünyesindeki iş grupları için ayrılabilirler.

Var sayılan kiracı konumlandırmalarında tüm yapılandırma var sayılan kiracı için gerçekleşir. Kiracı yöneticisi kullanıcıları ve grupları yönetir ve kiracıya yönelik marka, bildirim, iş politikalarını ve katalog arzlarını yapılandırır. Tüm kullanıcılar vRealize Automation konsoluna aynı URL ile bağlanırlar ama onlara sunulan yetenekler rolleri ile belirlenir.

Tek bir kiracı konumlandırmalarında sistem yöneticisi aynı vRealize Automation oluşumunu kullanan organizasyonu için tek bir yeni kiracı oluşturur. Kiracı kullanıcıları vRealize Automation konsoluna kiracıya özel bir URL’den bağlanırlar. Her ne kadar sistem çapında rolleri olan kullanıcılar her iki yapılandırmayı da görüp değişiklik yapabilseler de kiracı seviyesindeki yapılandırma var sayılan yapılandırmadan ayrılmıştır. Organizasyon kiracısı için IaaS yöneticisi fabrik gruplarını oluşturur ve fabrik yöneticisini atar. Fabrik yöneticisi organizasyon kiracısı içindeki iş grupları için rezervasyonlar oluşturabilir.

Çoklu kiracı konumlandırmalarında sistem yöneticisi aynı vRealize Automation oluşumunu kullanan her bir organizasyon için yeni kiracılar oluşturur. Her bir kiracı kullanıcısı vRealize Automation konsoluna o kiracıya özel URL’sinden bağlanırlar. Her ne kadar sistem çapında rolleri olan kullanıcılar her iki yapılandırmayı da görüp değişiklik yapabilseler de kiracı seviyesindeki yapılandırma var sayılan yapılandırmadan ayrılmıştır. Her bir kiracı için IaaS yöneticisi fabrik gruplarını oluşturur ve karşılık gelen fabriğe yöneticisini atar. Her ne kadar fabrik yöneticileri herhangi bir kiracı içindeki iş grubuna rezervasyonlar atayabilse de bu senaryoda genellikle sadece kendi kiracıları bünyesinde rezervasyonlar oluşturup yönetirler. Eğer aynı kimlik deposu birden çok kiracı için yapılandırılmışsa aynı kullanıcılar IaaS yöneticileri olarak ya da fabrik yöneticileri olarak atanabilirler.

Bu tasarımda iki ayrı iş grubuna sahip tek bir kiracı konumlandırılmaktadır. İlk iş grubu üretim iş yüklerinin provizyonlanması için, ikinci iş grubu ise geliştirme iş yüklerinin provizyonlanması için atanmışlardır.



Şekil 18 - İki Bölge için Kurum Bulut Otomasyonu Kiracı Tasarımı

Tablo 125 - Kiracı Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-BYP-024 | Ayrı kiracılar yerine ayrı iş grupları için vRealize Automation iş grupları kullanılacaktır | Ortamlar arası şeffaflık sağlar ve kaynaklar ve planlar gibi hizmetler için paylaşım sağlanmış olur. | Özellik grupları gibi bazı elemanlar her iki iş grubuna da görünebilir. Bu tasarım güvenlik ve denetim için izolasyon sağlamaz. |
| SDDC-BYP-025 | Her konumlandırma bölgesi için ayrı fabrik grupları oluşturulacaktır. Her fabrik grubu bölge özel veri merkezi kaynaklarını temsil eder. | İleride ihtiyaç duyulabilecek fabrik kaynaklarının izolasyonuna ve bağımsız fabrik yöneticilerine yetki devrine imkan sağlar. | İlk konumlandırma bir işlem podundan oluşan tek bir paylaşımlı fabrikten oluşur. |
| SDDC-BYP-026 | Kiracıların idaresi ve sistem çapında değişiklikler yapabilmek adına var sayılan kiracı ayarlarına sadece sistem yöneticisi erişecektir. | Var sayılan kiracı ayarları bireysel kiracı yapılandırmalarından izole edilmiş olur. | Her bir kiracı yöneticisi kendi kiracı yapılandırmasından sorumludur. |
| SDDC-BYP-027 | Dahili organizasyon yapısı ve iş yükü gereksinimleri gözden geçirilecektir. vRealize iş grupları, rezervasyonları, hizmet katalogları ve şablonları organizasyonun gereksinimlerine göre belirlenecektir. | vRealize Automation organizasyonun gereksinimlerine entegre olmak üzere tasarlanmıştır. Bu tasarımda hayali bir kurum başlangıç noktası olmak üzere seçilmiştir. | Müşteri ve iş ortakları iş gereksinimlerini birlikte değerlendirerek belirlemelidir. |

Hizmet kataloğu, IT hizmetlerinin kullanıcıları için ihtiyaç duydukları kaynakları ve istekte bulundukları ve yönetmek istedikleri hizmetler için ortak bir ara yüz sağlar. Bir kiracı yöneticisi ya da hizmet mimarı hizmet kataloğu hakkında, hizmet saatleri, destek takımı ya da değişiklik penceresi gibi bilgileri belirleyebilir. Her ne kadar hizmet kataloğu hizmetler üzerinde servis seviyesi anlaşmaları (SLA) zorunlu kılmasa da bu bilgiler kataloğu inceleyen iş kullanıcılarının erişimine açıktır.

Kullanıcılar istekte bulunmakta yetkili oldukları her katalog nesnesi için hizmet kataloğunu tarayabilirler. Bazı katalog nesneleri için istek, kullanıcının idare edebileceği bir nesnenin provizyonlanmasına yol açabilir. Örnek vermek gerekirse bir kullanıcı Windows Server 2012’nin önceden kurulmuş olan bir sanal makine için istekte bulunabilir ve sonrada bu sanal makineyi yönetebilir.

Kiracı yöneticileri yeni katalog nesneleri tanımlayabilir ve hizmet kataloğunda yayınlayabilirler. Kiracı yöneticisi akabinde katalog nesnelerinin tüketicilere sunumunu yönetebilir ve yeni nesneler sunabilir. Bir katalog nesnesini kullanıcıların erişimine açabilmek için kiracı yöneticisi nesneyi kendisine erişme yetkisi bulunan kullanıcı ya da gruplara yetkilendirebilir. Örnek vermek gerekirse bazı katalog nesneleri belirli iş grupları tarafından erişilebilirken diğerleri iş grupları arasında ortak olarak kullanılabilirler. Yönetici katalog nesnelerini farklı kullanıcılara, iş fonksiyonlarına bölümlerine veya fiziksel yerlerine göre atar.

Genellikle bir katalog nesnesi, provizyonlanacak kaynağın tam bir belirtimini içeren ve nesne için istekte bulunulduğunda işletilecek süreci sağlayan bir plan (blueprint) içerisinde tanımlanır. Bunun yanı sıra sanal makine özellikleri, kiralama süresi gibi nesnenin istemcisine açık olan seçenekleri tanımlar.

Bir makine planı sanal, bulut ya da fiziksel bir makinenin eksiksiz spesifikasyonlarıdır. Makine planı o makinenin özelliklerini, nasıl provizyonlandığını ve politika ve yönetim ayarlarını içerir. Makine planları hizmet kataloğu içinde katalog nesnesi olarak yayınlanırlar.

Makine planları bir iş grubuna özel ya da kiracı bünyesinde ortak paylaşılıyor olabilir. Kiracı yöneticileri herhangi bir iş grubunun kullanımı için paylaşımlı palnlar oluşturabilirler. İş grubu yöneticileri sadece belirli iş gruplarının kullanıcılarının erişebileceği planlar oluşturabilirler. Kiracı yöneticileri sadece iş grubu yöneticisi yetkilerine sahiplerse bu planları görüp üzerlerinde değişiklik yapıp silebilirler.

Bir kiracı yöneticisi bir planı paylaşımlı hale getirirse iş grubu yöneticisi paylaşılan planı başlangıç noktası olarak kullanıp kendi yeni planlarını oluşturabilir.

Tablo 126 - Tek Makine Planı

|  |  |
| --- | --- |
| Adı | Tanımı |
| Temel Windows Server (Geliştirme) | Geliştirme iş grubunun kullanımı için standart kurum Windows Server 2012 |
| Temel Windows Server (Üretim) | Üretim iş grubunun kullanımı için standart kurum Windows Server 2012 |
| Temel Linux (Geliştirme) | Geliştirme iş grubunun kullanımı için standart kurum Linux |
| Temel Linux (Üretim) | Üretim iş grubunun kullanımı için standart kurum Linux |
| Windows Server + SQL Server (Üretim) | Temel Windows 2012 R2 Server ile sessiz SQL 2012 özelleştirilmiş kurulumu, Üretim grubu için |
| Windows Server + SQL Server (Geliştirme) | Temel Windows 2012 R2 Server ile sessiz SQL 2012 özelleştirilmiş kurulumu, Geliştirme grubu için |

Aşağıda bulut platform konumlandırması içinde bulunan her bir hizmet tanımı gösterilmiştir:

Tablo 127- Temel Windows Server Gereksinimleri ve Standartları

|  |  |
| --- | --- |
| Hizmet Adı | Temel Windows Server |
| Provizyonlama Yöntemi | Kullanıcılar bu planı seçtiklerinde vRealize Automation önceden yapılandırılmış vCenter kişiselleştirmeleri ile bir vSphere sanal makine şablonunu klonlar. |
| Yetkili | Hem üretim hem de geliştirme iş grubu üyeleri. |
| Onay Süreci | Onaya gerek yoktur. Platforma erişilebilmesi onay sayılmaktadır. |
| İşletim Sistemi ve Sürüm Ayrıntıları | Windows Server 2012 R2 |
| Yapılandırma | Disk: tek bir disk sürücüsü, Ağ: Standart vSphere Ağı |
| Kira ve Arşiv Ayrıntıları | Üretim Planı: Son kullanım tarihi yok  Geliştirme Planı: En az 30 gün, en fazla 270 gün  Arşiv: 15 gün |
| Konumlandırma öncesi ve sonrası Gereksinimleri | Hizmet isteğini doğrulayan elektronik posta ayrıntıları ile yöneticiye iletilir. |

Tablo 128 - Temel Windows Plan Boyutlandırması

|  |  |  |  |
| --- | --- | --- | --- |
| Boyutlandırma | vCPU | Hafıza (GB) | Depolama (GB) |
| Var sayılan | 1 | 4 | 60 |
| Maksimum | 4 | 16 | 60 |

Tablo 129 - Temel Linux Server Gereksinimleri ve Standartları

|  |  |
| --- | --- |
| Hizmet Adı | Temel Linux Server |
| Provizyonlama Yöntemi | Kullanıcılar bu planı seçtiklerinde vRealize Automation önceden yapılandırılmış vCenter kişiselleştirmeleri ile bir vSphere sanal makine şablonunu klonlar. |
| Yetkili | Hem üretim hem de geliştirme iş grubu üyeleri. |
| Onay Süreci | Onaya gerek yoktur. Platforma erişilebilmesi onay sayılmaktadır. |
| İşletim Sistemi ve Sürüm Ayrıntıları | Red Hat Enterprise Server 6 |
| Yapılandırma | Disk: tek bir disk sürücüsü, Ağ: Standart vSphere Ağı |
| Kira ve Arşiv Ayrıntıları | Üretim Planı: Son kullanım tarihi yok  Geliştirme Planı: En az 30 gün, en fazla 270 gün  Arşiv: 15 gün |
| Konumlandırma öncesi ve sonrası Gereksinimleri | Hizmet isteğini doğrulayan elektronik posta ayrıntıları ile yöneticiye iletilir. |

Tablo 130 - Temel Linux Plan Boyutlandırması

|  |  |  |  |
| --- | --- | --- | --- |
| Boyutlandırma | vCPU | Hafıza (GB) | Depolama (GB) |
| Var sayılan | 1 | 6 | 20 |
| Maksimum | 4 | 12 | 20 |

Tablo 131 - Temel Windows Server ile SQL Server Kurulum Gereksinimleri ve Standartları

|  |  |
| --- | --- |
| Hizmet Adı | Temel Linux Server |
| Provizyonlama Yöntemi | Kullanıcılar bu planı seçtiklerinde vRealize Automation önceden yapılandırılmış vCenter kişiselleştirmeleri ile bir vSphere sanal makine şablonunu klonlar. |
| Yetkili | Hem üretim hem de geliştirme iş grubu üyeleri. |
| Onay Süreci | Onaya gerek yoktur. Platforma erişilebilmesi onay sayılmaktadır. |
| İşletim Sistemi ve Sürüm Ayrıntıları | Windows Server 2012 R2 |
| Yapılandırma | Disk: tek bir disk sürücüsü, Ağ: Standart vSphere Ağı  Sessiz Kurulum: Plan vRealize Automation ajanı kullanarak sessiz bir betik çalıştırarak SQL 2012 veri tabanını kişiselleştirilmiş ayarlar ile kurar. |
| Kira ve Arşiv Ayrıntıları | Üretim Planı: Son kullanım tarihi yok  Geliştirme Planı: En az 30 gün, en fazla 270 gün  Arşiv: 15 gün |
| Konumlandırma öncesi ve sonrası Gereksinimleri | Hizmet isteğini doğrulayan elektronik posta ayrıntıları ile yöneticiye iletilir. |

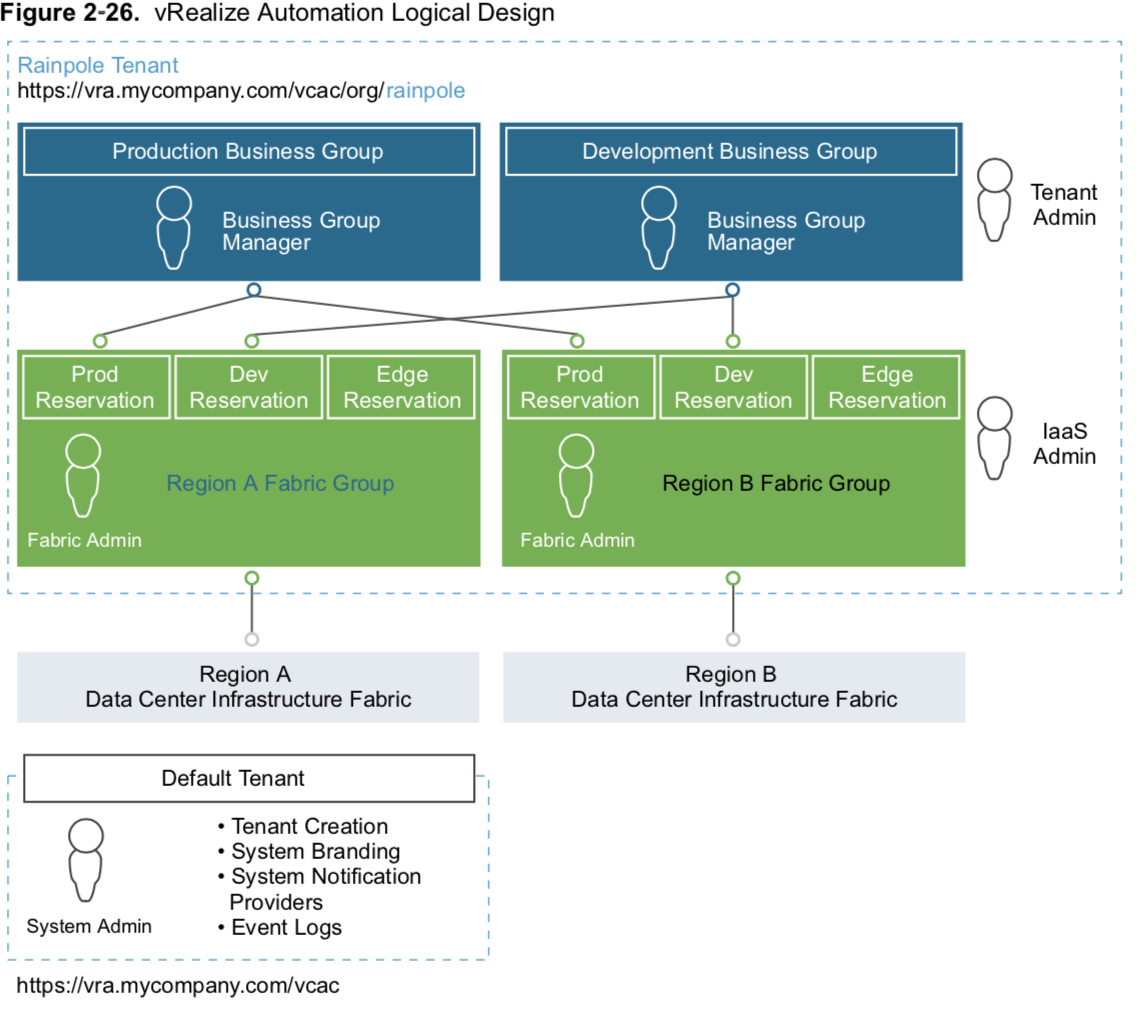
Tablo 132 - Temel Windows ile SQL Server Plan Boyutlandırmaları

|  |  |  |  |
| --- | --- | --- | --- |
| Boyutlandırma | vCPU | Hafıza (GB) | Depolama (GB) |
| Var sayılan | 1 | 8 | 100 |
| Maksimum | 4 | 16 | 100 |

Sistem yöneticileri vRealize Automation konsol görünüşünü logo, arka plan rengi veya başlık ve dipnotları değiştirmek suretiyle site özelinde markalaşma hedeflerine ulaşabilirler. Kiracı yöneticileri ise ister var sayılan ister kiracının markasına özel yapılandırmaları kullanabilirler.

* + - 1. vRealize Automation Hizmet Olarak Alt Yapı Tasarımı

Bu başlık altında vRealize Automation’ın vSphere kaynakları ile entegre olarak SDDC ile kullanılmak üzere hizmet olarak alt yapı tasarımı sunulmaktadır. Aşağıdaki şekilde vRealize Automation gruplarını ve vSphere kaynaklarının mantıksal tasarımını gösterilmektedir.

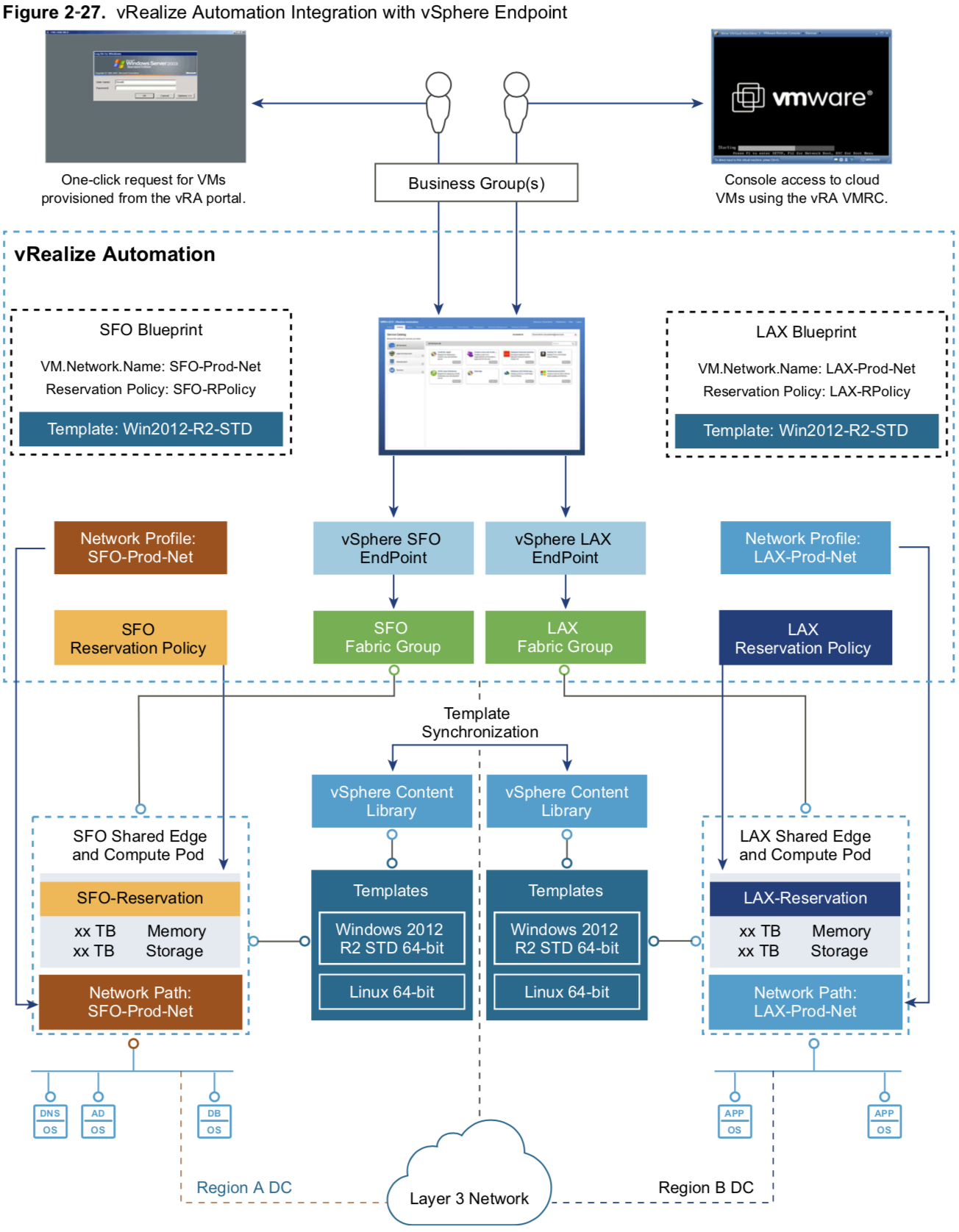


Şekil 19 - vRealize Automation Mantıksal Tasarımı

Aşağıdaki terimler vRealize Automation, vSphere ile entegre olduğunda uygulanabilirler. Bu terimler ve anlamları sadece vSphere söz konusu olduğundan farklı olarak değişiklik gösterebilirler.

|  |  |
| --- | --- |
| Terim | Tanımı |
| vSphere (vCenter Server) uç noktası | vRealize Automation IaaS’ın vSphere işlem kaynaklarına erişmek için gereken bilgileri sağlar. |
| İşlem Kaynağı | vRealize Automation içerisindeki bir vCenter Server kümesini ya da kaynak havuzunu ya da datastore veya datastore kümesini temsil eden sanal nesnelerdir. |
| Fabrik Grupları | vRealize Automation IaaS işlem kaynaklarını fabrik gruplarına ayırarak organize eder. |
| Fabrik Yöneticisi | Fabrik yöneticileri fabrik grupları olarak organize edilmiş işlem kaynaklarını yönetirler. |
| İşlem Rezervasyonu | Sanal makinelerin provizyonlanması için belirli iş gruplarının kullanımına rezerve edilen CPU ve hafıza gibi işlem kaynaklarının hissesi. |
| Depolama Rezervasyonu | İşlem rezervasyonlarına benzer fakat sadece mevcut depolama hisselerine ait rezervasyonlardır. Bu kapsamda gigabyte cinsinden mevcut LUN ya da datastore dan verilir. |
| İş Grupları | Sanal makine tüketicilerinin bir koleksiyonu olup genellikle bir organizasyonun iş birimlerine ya da departmanlarına karşılık gelirler. Sadece iş gruplarındaki kullanıcılar sanal makine talep edebilirler. |
| Rezervasyon Politikası | vRealize Automation IaaS rezervasyonu (aynı zamanda sanal rezervasyon da denir) belirli bir provizyonlanmış sanal makineden türetir. Rezervasyon politikası mantıksal bir etiket ya da orijinal rezervasyonu gösteren bir işarettir. |
| Plan (Blueprint) | Bir sanal makinenin özelliklerini, nasıl provizyonlanacağını ve politika ve yönetim ayarlarını belirleyen bir belirtimdir. Planlar aynı zamanda bir iş grubu kullanıcılarının sanal bir rezervasyon üzerinde (işlem kaynağı) rezervasyon politikasına uygun olarak sanal makineler oluşturabilmesine yararlar. |

Aşağıdaki şekilde bu bölümde tarif edilen veri merkezleri arasında provizyonlanmış bir vSphere’a entegre vRealize Automation kurulumu gösterilmektedir:



Şekil 20 - vRealize Automation ile vSphere Uç Nokta Entegrasyonu

Bir alt yapı kaynağı uç noktası bir ya da birden çok kaynak setini sağlayan alt yapıya yapılan bağlantılardır. Bu bağlantılar IaaS yöneticisi tarafından son kullanıcıların tüketimi için erişilebilir hale getirilirler. vRealize Automation IaaS düzenli olarak bu bilinen uç nokta kaynaklarından ve onlar içerisinde barınan sanal kaynaklarla ilgili bilgiler toplar. Uç nokta kaynaklarına işlem kaynakları da (ya da işlem podu) denmektedir.

Alt yapı verileri uç nokta kaynakları ile iletişimde olan ve onları yöneten proxy ajanları vasıtasıyla toplanırlar. Her bir alt yapı uç noktasındaki işlem kaynakları ve her işlem kaynağı üzerinde provizyonlanmış makineler ile ilgili bilgiler düzenli olarak toplanırlar.

vRealize Automation IaaS bileşenlerinin kurulumları esnasında proxy ajanları ve onlara tanımlı uç noktaları yapılandırılabilirler. Buna alternatif olarak kurulum sonrasında da proxy ajanları ve uş noktaları tanımlanarak yapılandırılabilirler.

Tablo 133 - Uç Nokta Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-BYP-028 | İki adet vSphere uç noktası oluşturulacaktır. | Her bölgedeki he bir vCenter Server oluşumu için bir vSphere uç noktası gereklidir. İki bölge için iki uç nokta gerekir. | Yeni bölgeler eklendikçe ek vSphere uç noktaları konumlandırılmaları gereklidir. |
| SDDC-BYP-029 | Gömülü vRealize Orchestrator oluşumuna bağlanmak için yapılandırılmış bir vRealize Orchestrator uç noktası oluşturulacaktır. | vRealize Automation uzantısı olarak vRealize Orchestrator’ı kullanmaktadır. Bu tasarımda bir gömülü Orchestrator olduğundan bir adet uç noktası yeterlidir. | vRealize Orchestrator uç noktasının yapılandırılmasını gerektirir |
| SDDC-BYP-030 | vSphere uç noktası ile ilişkilendirilecek bir NSX uç noktası oluşturulacaktır. | NSX uç noktaları NSX yöneticisine bağlanmak için ve tüm NSX ile alakalı vRealize Automation planlarında. Bulunan operasyonların etkinleşebilmesi için gereklidir | Yoktur. |

Bir sanallaştırma işlem kaynağı, bir ESXi sanallaştırma sunucusunu veya sunucular kümesini temsil eden bir vRealize Automation nesnesidir. Bir grup üyesi bir sanal makine için istekte bulunduğunda, sanal makine bu işlem kaynakları üzerinde provizyonlanır. vRealize Automation düzenli olarak bilinen işlem kaynakları ve onlar üzerinde proviyonlanmış sanal makineler hakkında proxy ajanları vasıtasıyla bilgi toplar.

Var sayılanda işlem kaynakları işlem kümesinin kökünde provizyonlanırlar. Bu tasarımda vSphere kaynak havuzlarının kullanılması zorunludur.

Tablo 134 - İşlem Kaynakları Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-BYP-031 | Her bir konumlandırılmış bölge için en az bir işlem kaynağı oluşturulacaktır. | Her bölgede bir işlem kümesi bulunur, bir işlem kaynağı her bir küme için gereklidir. | Ek işlem kümeleri oluşturuldukça o bölgede bulunan mevcut işlem kaynaklarına eklenmelidirler ya da yeni bir tanesi oluşturulmalıdır. |

Bir fabrik grup çeşitli işlem kaynaklarının oluşturduğu mantıksal konteynırlardır ve fabrik yöneticileri tarafından yönetilirler.

Tablo 135 - Fabrik Grup Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-BYP-032 | Her bölge için bir fabrik grubu oluşturulacak ve tüm o bölgedeki işlem kaynakları ve kenar kaynakları bu grup içerisinde olacaklardır. | Bölgeye özel provizyonlama yapılabilmesi için her bölge için bir fabrik grup oluşturulması gereklidir. | Bir bölgeye ek kümeler eklendikçe fabrik gruba eklenmelidirler. |

Bir iş grubu genellikle bir iş departmanına ya da başka türlü organizasyonel bir birime karşılık gelen makine tüketicilerinin bir koleksiyonudur. Bir makine için istekte bulunabilmesi için bir vRealize Automation kullanıcısının en az bir iş grubuna üye olması gereklidir. Her grubun makine isteğinde bulunabilmesi kullanılan bir set plana erişimi vardır. İş grupları aşağıdaki özelliklere sahiptirler:

* Bir grubun en az bir grup yöneticisi olmalı ve bu yönetici planları idame ettirmeli ve makine isteklerini onaylamalıdır.
* Gruplarda diğer grup üyeleri adına makine isteğinde bulunabilip o makineleri yönetebilen destek kullanıcıları olabilir.
* Bir vRealize Automation kullanıcısı bir veya birden fazla iş grubuna üye olabilir ve farklı gruplarda farklı rollere sahip olabilirler.

Bir rezervasyon belirli bir fabrik grubu tarafından kullanılmak için ayrılmış işlem kaynağının mevcut hafıza, işlemci ve depolama hisseleridir. Her rezervasyon tek bir fabrik grubu içindir fakat ilişki çokludan çokluyadır. Bir fabrik grubunun bir işlem kaynağı üzerinde birden fazla rezervasyonu olabileceği gibi birden fazla işlem kaynağında rezervasyonları da olabilir. Hatta her ikisi de aynı anda doğru olabilir.

Her ne kadar rezervasyonlar vRealize Automation içinde küme hafızasının ya da depolamasının bir bölümünün tahsis edilmesi için bir yöntem sunsa da rezervasyonlar altta yatan vSphere işlem kaynaklarında bir sıkışıklık olduğunda işlemci ve hafızanın nasıl tahsis edileceğini kontrol etmezler. Bu sıkışıklık durumlarında tahsislerin kontrol edilebilmesi için vSphere kaynak havuzları kullanılmalıdır. Bu özellikten tamamıyla faydalanabilmek için tüm sanal makineler 4 kaynak havuzundan bir tanesine yerleştirilmelidirler: ank01-w01rp-sddc-edge, ank01-w01rp-sddc-mgmt, ank01-w01rp-user-edge ve ank01-w01rp-user-vm.

Kaynak havuzlarının ayrıntıları aşağıdaki gibidir:

* ank01-w01rp-sddc-edge veri merkezi seviyesinde NSX kenar bileşenleri için atanmıştır ve herhangi bir kullanıcı iş yükü barındırmamalıdırlar.
* ank01-w01rp-sddc-mgmt veri merkezindeki sanal makinelerin yönetimi için atanmıştır.
* ank01-w01rp-user-edge NSX kenar geçitleri veya belirli müşteri iş yüklerine hizmet eden yük dengeleyicileri gibi statik ya da dinamik olarak konumlandırılmış NSX bileşenlerine atanmıştır.
* Ank01-w01rp-user-vm belirli müşteri iş yüklerini barındıran Windows, Linux, veri tabanları gibi statik ya da dinamik olarak konumlandırılmış sanal makinelere atanmıştır.

Tablo 136 - Rezervasyon Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-BYP-033 | Her bölgede her bir iş grubu için en az bir vRealize Automation rezervasyonu oluşturulacaktır. | Verilen örnekte her kaynak kümesinin birisi üretim birisi geliştirme için olmak üzere iki adet rezervasyonu vardır. | Üretim ve geliştirme aynı işlem kaynaklarını paylaştıkları için geliştirme iş grubu sabit miktarda kaynaklarla sınırlandırılmalıdırlar. |
| SDDC-BYP-034 | Her bölgedeki kenar kaynakları için en az bir vRealize Automation rezervasyonu oluşturulacaktır. | Her bölgedeki kenar rezervasyonları NSX’in istek üzerine kenar hizmet geçitleri oluşturup kenar kümesine yerleştirebilmesine izin verir. | İş yük rezervasyonu kenar rezervasyonunu ağ ayarlarında tanımlamalıdır. |
| SDDC-BYP-035 | Tüm vRealize Automation iş yükleri ank01-w01rp-user-vm kaynak havuzunu kullanacak şekilde ayarlanacaktır. | NSX ağ bileşenleri için atanmış işlem kaynaklarını garantiye alabilmek için son kullanıcı tarafından konumlandırılmış iş yükleri atanmış bir son kullanıcı vSphere kaynak havuzuna konulmalıdır. Kök seviyede konumlandırılmış iş yükleri kaynak havuzlarındaki iş yüklerine kıyasla daha fazla kaynağa sahip olacaklarından sıkışıklık yaratabileceklerdir. | Bulut yöneticileri tüm iş yükü rezervasyonlarının uygun kaynak havuzu için yapılandırıldığından emin olmalıdır. |
| SDDC-BYP-036 | Dinamik olarak provizyonlanan NSX kenar bileşenlerinin ank01-w01rp-user-edge kaynak havuzunu kullanacak şekilde vRealize Automation yapılandırılacaktır. | NSX ağ bileşenleri için atanmış işlem kaynaklarını garantiye alabilmek için son kullanıcı tarafından konumlandırılmış NSX kenar bileşenleri atanmış bir son kullanıcı vSphere kaynak havuzuna konulmalıdır. Kök seviyede konumlandırılmış iş yükleri kaynak havuzlarındaki iş yüklerine kıyasla daha fazla kaynağa sahip olacaklarından sıkışıklık yaratabileceklerdir | Bulut yöneticileri tüm iş yükü rezervasyonlarının uygun kaynak havuzu için yapılandırıldığından emin olmalıdır. |
| SDDC\_BYP-037 | Kenar veya işlem iş yükleri için kullanılan tüm vCenter kaynak havuzları kök seviyesinde tanımlanmalıdırlar. İç içe geçmiş kaynak havuzları tavsiye edilmemektedir. | İç içe geçmiş kaynak havuzları yönetimsel olarak kompleks kaynak hesaplamalarına yol açabilerek istenmeyen fazla ya da az kaynak atanmasına yol açabilir. | Tüm kaynak havuzları kök seviyede oluşturulmalıdır. |

Her bir sanal rezervasyonu bir rezervasyon politikasına eklenebilir. Provizyonlanmış belirli bir sanal makinenin rezervasyonu vRealize Automation tarafından öncelikler, fabrik grubunun rezervasyonlarının mevcut kullanımı ve diğer kişisel özellikler göz önüne alınarak plan içerisinde tanımlanmış rezervasyon politikasına göre belirlenir.

Tablo 137 - Rezervasyon Politikaları Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-BYP-038 | Her bölge için en az bir iş yükü rezervasyonu politikası oluşturulacaktır. | Rezervasyon politikaları her bölgedeki belirli bir rezervasyon setine işaret etmek için kullanılırlar. Aynı zamanda uygun bölge, işlem kümesi ve veya vSphere kaynak havuzuna hedeflerler. | Yoktur. |
| SDDC-BYP-039 | Kenar kümelerinde konumlandırılacak dinamik olarak oluşturulan kenar hizmet geçitleri için en az bir rezervasyon politikası oluşturulacaktır. | Kenar cihazların karşılık gelen kenar kümesine ve/veya vSphere kaynak havuzuna yerleştirilebilmeleri için gereklidir. | Yoktur. |

Bir veri depolama rezervasyon politikası disk provizyonlanmasını sınırlayan bir makine planına atanmış bir set datastore’lardır. Veri depolama rezervasyon politikaları oluşturulur ve uygun datastore’lara ilişkilendirilir ve rezervasyonlara atanırlar.

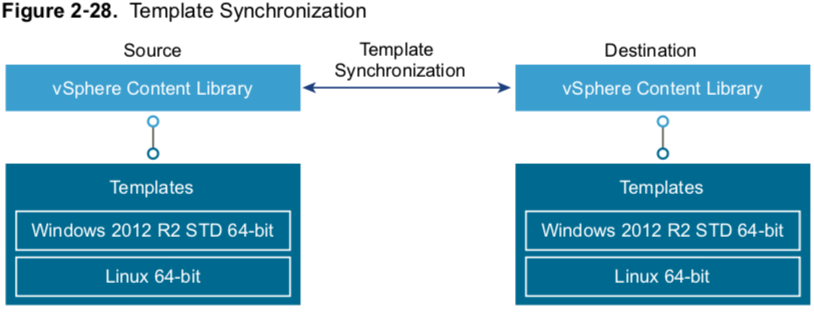
Tablo 138 - Depolama Rezervasyon Politika Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-BYP-040 | Bu tasarım içerisinde depolama katmanları kullanılmayacaktır. | Altta yatan fiziksel veri depolama tasarımı depolama katmanlarını kullanmamaktadır. | Her iki iş grubunun da aynı veri depolamaya erişimi vardır. Farklı depolama özelliklerine sahip birden fazla datastore kullanan müşteriler için vRealize Automation depolama rezervasyon politikaları değerlendirilmelidir. |

Bu çift bölgeli tasarım aynı tek makine planlarını kullanan aynı portal vasıtasıyla iş yüklerinin bölgeler arası provizyonlanmasını desteklemektedir. Bölgeler arasında tutarlı şablonlara sahip olunması için bir senkronizasyon mekanizması gereklidir. Bu senkronizasyonun sağlanması için ise vSphere Content Library ya da vCloud Connector ve vSphere Replication gibi harici hizmetlerden birisi kullanılabilir.

Tablo 139 - Şablon Senkronizasyon Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-BYP-041 | Bu tasarımda şablonların bölgeler arasında senkronizasyonu için vSphere Content Library kullanılacaktır. | vSphere Content Library kullanılacak vSphere sürümü içerisinde hali hazırda mevcuttur ve şablonların senkronizasyonu için gereksinimleri karşılamaktadır. | Depolama alanları her bölgede provizyonlanmış olmalıdır. vRealize Automation vSphere Content Library den şablonları direk olarak tüketemez. |



Şekil 21 - Şablon Senkronizasyonu

VMware Identity Manager kiracı kimlik yönetimi yapar ve vRealize Automation cihazı içerisinde entegredir. Identity Manager kurum active directory etki alanı ile senkronize olur. Önemli kullanıcılar ve gruplar Identity Manager ile senkronize edilirler. Kimlik doğrulama active directory etki alanı kullanılarak yapılmakta ama aramalar vRealize Automation üzerindeki yerel active directory kopyası üzerinde yapılmaktadır.



Şekil 22 - VMware Identity Manager ile Active Directory arasında Kimlik Doğrulama Proxylenmesi

Tablo 140 - Active Directory Kimlik Doğrulama Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-BYP-042 | Directory Service seçeneği olarak “Active Directory with Integrated Windows Authenticaton” seçilecektir. | Kurum tek forest, çoklu etki alanı active directory ortamına sahiptir. Entegre Windows kimlik doğrulaması çoklu etki alanı veya çoklu forest active directory ortamları arasında güven sağlamayı desteklemektedir. | vRealize Automation cihazının active directory etki alanına katılması gerekmektedir. |

Var sayılanda vRealize Automation cihazı küçük active directory ortamlarını desteklemek için yeterli olan 18GB hafıza ile yapılandırılmaktadır. Bir active directory ortamı organizasyonel birimler içerisinde 25.000’den daha az kullanıcı varsa küçük olarak nitelendirilmektedir. Daha çok kullanıcıya sahip active directory ortamları ek hafıza ve işlemci kaynağı gerektirir.

Birleştirici bileşeni vRealize Automation hizmeti ile active directory arasında kullanıcı ve grupların senkronizasyonunu sağlamaktadır. Bununla beraber birleştirici varsayılan kimlik sağlayıcıdır ve kullanıcıları hizmete kimlik doğrulama yaparak bağlayandır.

Tablo 141 - Birleştirici Yapılandırma Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-BYP-043 | Directory hizmetlerinin yüksek erişilebilirliğini destekleyebilmek için ikinci bir birleştirici yapılandırılacak ve ikinci bir vRealize Automation cihazına karşılık gelecektir. | Bu tasarım yüksek erişilebilirliği iki adet vRealize Automation cihazı kurarak ve yük dengelemeli NSX kenar oluşumları ile sağlanmaktadır. İkinci bir birleştiricinin ikinci vRealize Automation’a bağlayarak yedeklilik sağlanmakta ve performans yün dengeleme ile artırılmaktadır. | Bu tasarım mevcut kurulu HA yeteneklerinin kullanarak kurulumu basitleştirmektedir. Tasarımda yük dengelemede NSX yük dengeleyicisini kullanmaktadır. |

* + 1. vRealize Business for Cloud Tasarımı

vRealize Business for Cloud son kullanıcılara iş yüklerinin yürütülmesi ile ilintili maliyetlerin şeffaf bir şekilde yansıtılmasına yarar. vRealize Business gibi bir sistem ile iş yüklerinin finansal maliyetlerinin toplanması ve bir araya getirilmesi SDDC’nin sağladığı değerin gösterilmesinde faydalıdır.

vRealize Business vRealize Automation ile entegre olarak iş yükü istekleri esnasında maliyeti gösterir ve kullanıcı, iş grubu ya da kiracı bazında süregelen maliyetleri yansıtabilir. Ek olarak kiracı yöneticileri organizasyonlarının gereksinimleri çerçevesinde özelleştirilmiş raporlar oluşturabilirler.

Tablo 142 - vRealize Business for Cloud Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-BYP-044 | Bulut yönetim platformu kapsamında vRealize Business for Cloud konumlandırılacaktır ve vRealize Automation ile entegre edilecektir. | Kiracı ve iş yükü maliyetleri vRealize Business for Cloud tarafından sağlanmaktadır. | vRealize Business for Cloud ve uzak kollektörleri için ek cihazları konumlandırmak gerekir. |
| SDDC-BYP-045 | Varsayılan vRealize Business for Cloud boyutu (8GB) kullanılacaktır. Uzak kollektörler içinse azaltılmış (2GB) hafıza kullanılacaktır. | Varsayılan vRealize Business for Cloud cihazı 10.000 sanal makineye kadar desteklemektedir. Uzak kollektörler sunucu servisleri çalıştırmayacaktır ve 2GB hafıza ile çalışabilmektedirler. | Yoktur. |
| SDDC-BYP-046 | Varsayılan vRealize Business referans maliyet ver tabanı kullanılacaktır. | Varsayılan referans maliyetler endüstri bilgilerine dayanmaktadır ve düzenli olarak güncellenmektedir. | Varsayılan referans maliyetler gerçek müşteri maliyetlerini tam olarak yansıtmayabilir. vRealize Business cihazı referans veri tabanını güncelleyebilmek için Internet bağlantısına ihtiyaç duymaktadır. |
| SDDC-BYP-047 | vRealize Business üçlü sanal makine mimarisi olarak, a ve b bölgesinde uzak kolektörleri olacak şekilde kurulacaktır. | En iyi performansın alınabilmesi için vRealize Business kollektörleri bölgesel olarak veri toplayacağı kaynağın yerelinde bulunmalıdırlar. Bu tasarım felaket kurtarmayı kapsadığından bulut yönetim platformu a veya b bölgesinde de bulunabilir. | Felaket kurtarma senaryosunun geçerli olmadığı ortamlarda ek bir cihaz daha konumlandırılması gerekir. Her ne kadar vRealize Business sunucusu kendi başına yükü kaldırabilse de uzak veri kollektörü için ek bir cihaz gereklidir. |
| SDDC-BYP-048 | vRealize Business sunucu sanal makinesi bölgeler arası mantıksal ağında yer alacaktır. | vRealize Business konumlandırması vRealize Automation’a dayanmaktadır. Bir felaket esnasında vRealize Business vRealize Automation ile birlikte göç edecektir. | Yoktur. |
| SDDC-BYP-049 | Her bir bölge spesifik mantıksal ağda bir vRealize Business uzak veri kolleltör sanal makinesi konumlandırılacaktır. | vRealize Business uzak veri kollektörü bölge spesifik bir kurulum gerektirir. Felaket esnasında uzak kollektör vRealize Automation ile göç etmeye ihtiyaç duymaz. | vCenter Server ile olan iletişim bir NSX kenar cihazından geçen ek bir Layer 3 atlama gerektirir. |

Tablo 143 - vRealize Business for Cloud Sanal Cihaz Kaynak Gereksinimleri

|  |  |
| --- | --- |
| Özellik | Değer |
| vCPU sayısı | 4 |
| Hafıza | Sunucu için 8 GB / Kollektör için 2 GB |
| vRealize Business Fonksiyonu | Sunucu ya da Uzak Kollektör |

* + 1. vRealize Orchestrator Tasarımı

VMware vRealize Orchestrator bir geliştirme ve süreç otomasyon platformudur. Bu platform genişleyebilir bir iş akışı kütüphanesi sağlar ve bu iş akışları ile vSphere alt yapısı, diğer VMware ve üçüncü parti teknolojilerini yönetebilen yapılandırılabilir süreçler oluşturulup koşturulabilir.

Bu tasarımda vRealize Automation işlem kaynak atamaları için vCenter Server’a vRealize Orchestrator eklentisi ile bağlanır.

* + - 1. vRealize Orchestrator Mantıksal Tasarımı

Bu tasarım atanmış veya harici bir vRealize Orchestrator oluşumu yerine vRealize Automation cihazı içerisine gömülü olan Orchestrator oluşumunu kullanmaktadır.

Tablo 144 - vRealize Orchestrator Donanım Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-BYP-VRO-001 | vRealize Automation oluşumu içinde gömülü bulunan vRealize Orchestrator kullanılacaktır. | Gömülü vRealize Orchestrator kullanılmasının daha hızlı konumlandırma, yönetilecek cihaz sayısının azalması, sürüm yükseltilmesinin kolaylaşması, performans iyileşmesi ve harici bir veri tabanı ihtiyacı olmaması avantajları bulunmaktadır. | Tasarım bütününü basitleştirerek daha az sayıda cihaz konumlandırılmak zorunda kalınır. |

vRealize Orchestrator kimlik doğrulamak için sadece vRealize Automation kimlik doğrulamasını desteklemektedir.

Tablo 145 - vRealize Orchestrator Kimlik Hizmetleri Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-BYP-VRO-002 | Gömülü vRealize Orchestrator vRealize Automation kimlik doğrulamasını kullanacaktır | Yegâne kimlik doğrulama yöntemi budur. | Yoktur. |
| SDDC-BYP-VRO-003 | vRealize Orchestrator vRealize Automation kiracı kurumu kullanıcılarını kimlik doğrulama için kullanacaktır. | vRealize Automation var sayılan kiracı kullanıcıları sadece yöneticilerdir. Müşteri kiracısına bağlanarak iş akışlarının son kullanıcılara verilmiş yetkiler ile çalışmaları sağlanır. | vRealize Orchestrator iş akışları işletecek son kullanıcıların gerekli yetkilere sahip olmaları gerekmektedir. Bazı eklentiler vRealize Automation kimlik doğrulaması ile doğru çalışmayabilirler. |
| SDDC-BYP-VRO-004 | Bir vRealize Orchestrator kurulumu sadece bir müşteri kiracısı ile ilişkilendirilecektir. | Potansiyel kiracılar arasında ayrım ve mümkün olan en iyi seviyede güvenlik sağlamak için vRealize Orchestrator kurulumu tek bir kiracı ile ilişkilendirilmelidir. | Eğer ek vRealize Automation kiracıları yapılandırılırsa ek Orchestrator kurulumları da gerekir. |

vRealize Orchestrator diğer sistemlerle iletişim sağlamak için belirli ağ portları kullanır. Bu portlar varsayılan ayarları ile bırakılabilir ya da istenen değişiklikler yapılabilir. Eğer bu portlarda değişiklik yapılırsa bu portlara olan erişimlerin kontrol edilmesi gerekir. Özelleştirilmiş eklentiler kullanılıyorsa diğer başka portlara da ihtiyaç duyulabilir.

Tablo 146 - vRealize Orchestrator Varsayılan Yapılandırma Portları

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Port | Numarası | Protokol | Kaynak | Hedef | Tanımı |
| HTTPS sunucu Portu | 443 | TCP | Son kullanıcı Web browser | Gömülü vRealize Orchestrator sunucusu | vRealize Orchestrator REST API’sine SSL ile güvenli http protokolü ile bağlantı |
| Web yapılandırma HTTPS erişim portu | 8283 | TCP | Son kullanıcı web browser | vRealize Orchestrator yapılandırması | vRealize Orchestrator yapılandırması için web ara yüzü. |

vRealize Orchestrator’ün harici servisler ile iletişimi için güvenlik duvarında harici ağ portları için kural tanımlanması gereklidir.

Tablo 147 - vRealize Orchestrator Varsayılan Harici İletişim Portları

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Port | Numarası | Protokol | Kaynak | Hedef | Tanımı |
| LDAP | 389 | TCP | vRealize Orchestrator Sunucusu | LDAP sunucusu | LDAP kimlik doğrulama sunucu sorguları |
| SSL ile LDAP | 636 | TCP | vRealize Orchestrator Sunucusu | LDAP sunucusu | Güvenli LDAP kimlik doğrulama sorguları |
| Global Katalog kullanan LDAP | 3268 | TCP | vRealize Orchestrator sunucusu | Global Katalog sunucusu | Microsoft Global Katalog sunucusuna yönlendirilen sorgular |
| DNS | 53 | TCP | vRealize Orchestrator sunucusu | DNS sunucusu | İsim çözümleme |
| vCenter SSO | 7444 | TCP | vRealize Orchestrator sunucusu | vCenter SSO sunucusu | vCenter Single Sign On sunucusu ile iletişim portu |
| SQL Server | 1433 | TCP | vRealize Orchestrator sunucusu | Microsoft SQL sunucusu | vRealize Orchestrator veri tabanı olarak yapılandırılmış Microsoft SQL Server ya da Express için erişim portu |
| PostgreSQL | 5432 | TCP | vRealize Orchestrator sunucusu | PostgreSQL sunucusu | vRealize Orchestrator veri tabanı olarak yapılandırılmış PostgreSQL için erişim portu |
| Oracle | 1521 | TCP | vRealize Orchestrator sunucusu | Oracle DB sunucusu | vRealize Orchestrator veri tabanı olarak yapılandırılmış Oracle DB için erişim portu |
| SMTP Server | 25 | TCP | vRealize Orchestrator sunucusu | SMTP sunucusu | Elektronik posta duyuruları için kullanılan port |
| vCenter Server API | 443 | TCP | vRealize Orchestrator sunucusu | vCenter Server | vCenter Server API portu |
| vCenter Server | 80 | TCP | vRealize Orchestrator sunucusu | vCenter Server | HTTPS tüneli için kullanılan port |
| VMware ESXi | 443 | TCP | vRealize Orchestrator sunucusu | ESXi sanallaştırma sunucuları | Seçimlik olarak iş akışları ESXi sanallaştırma sunucuları üzerinden de yapılabilir |

vRealize Orchestrator bağımsız ya da küme kipinde çalışmayı desteklemektedir. Bu tasarım küme kipini kullanmaktadır. vRealize Orchestrator sunucusunun bağımsız kipi varsayılan ayardır. vRealize Orchestrator hizmetlerinin erişilebilirliğinin artırılması için ve daha yüksek erişilebilirliğe sahip SDDC oluşturabilmek için küme kipinde yapılandırılması gereklidir. Birden çok vRealize Orchestrator oluşumu paylaşımlı bir veri tabanı ile çalıştırılabilir. Küme kipinde vRealize Orchestrator oluşumları özdeş sunucu ve eklenti yapılandırmaları ile bir arada çalışmaktadır.

Tüm vRealize Orchestrator oluşumları birbirleriyle belirli aralıklarla kullanılabilirlik sinyali göndermek suretiyle konuşurlar. Sadece aktif vRealize Orchestrator oluşumu istemci isteklerine cevap vererek iş akışlarını çalıştırırlar. Eğer aktif sunucu kullanılabilirlik sinyali gönderemezse pasif oluşum görevleri devralarak iş akışlarını kesildikleri yerden devam ettirir. Kullanılabilirlik sinyali paylaşımlı veri tabanı aracılığıyla uygulanır bu sebeple bir vRealize Orchestrator kümesinin ağ tasarımına bir etkisi yoktur.

vRealize Automation oluşumları içerisinde gömülü vRealize Orchestrator’lar arasında yük dengelemesi yapılandırılarak vRealize Orchestrator kontrol merkezine ağ erişiminin sağlanmalıdır.

Tablo 148 - vRealize Orchestrator SDDC Küme Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-BYP-VRO-005 | Gömülü vRealize Orchestrator kontrol merkezine ağ erişimi yük dengeleyici aracılığıyla sağlanacaktır. | Kontrol merkezi vRealize Orchestrator’un kiracı yapılandırması ve sertifikalar gibi özelleştirilmelerine izin verir. Yük dengeleyici ile erişim ile iki pod tasarımına genişletmek mümkün olur. | Yoktur. |

vRealize Orchestrator’un vCenter sunucusuna bağlanırken kimlik doğrulama ve yetkilendirmesi için bir servis hesabı kullanılmalıdır. Bu hesap ile SDDC içindeki sanal nesnelerin oluşturulması ve vCenter’ın orkestrasyonu gerçekleştirilirler.

Tablo 149 - Kimlik Doğrulama ve Yetkilendirme Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-BYP-VRO-006 | vRealize Orchestrator ile vSphere arasındaki uygulamadan uygulamaya iletişim için vCenter üzerinde svc-vro adlı bir servis hesabı yapılandırılacaktır. | SDDC bileşenleri arasında istek cevap etkileşimlerinin takibi açısından izlenebilirlik artar. | Servis hesabının yaşam döngüsünün SDDC dışında takip edilmesi gerekmektedir. |
| SDDC-BYP-VRO-007 | vCenter Server içinde svc-vro servis hesabı oluşturulurken yerel yetkiler kullanılacaktır. | Yerel yetkilerin kullanılması sadece işlem vCenter server oluşumlarının geçerli ve vRealize Orchestrator tarafından erişilebilir uç noktalar olduğunu garantiler. | Ek işlem vCenter Server konumlandırılırsa servis hesabının o vCenter server üzerinde de yerel yetkilere sahip olması gerekir. |

* + - 1. vRealize Orchestrator Yapılandırması

vRealize Orchestrator yapılandırması istemci ve veri tabanı yapılandırması, SSL sertifikalar ve eklentiler hakkında yönlendirmeleri içerir.

vRealize Orchestrator istemcisi paketlerin içeri aktarılmasını, iş akışlarının oluşturulması, çalıştırılması, tarifelendirilmesi ve kullanıcı izinlerinin yönetilmesini sağlayan bir masa üstü uygulamadır. İstemci bir masa üstü sistemine bağımsız olarak kurulabileceği gibi Java WebStart kullanılarak vRealize Automation cihazının konsolundan direk olarak da çalıştırılabilir.

vRealize Orchestrator yapılandırma ara yüzü vCenter Server, ilişkisel ver tabanı yönetim sistemi (RDBMS), LDAP, vCenter SSO ve diğer sunucular ile güvenli bağlantılar ile iletişim kurar. İhtiyaç olan SSL sertifikası bir URL veya dosyadan yüklenebilir. vCenter Server SSL sertifikası vRealize Orchestrator yapılandırma ara yüzünde SSL Trust Manager sekmesinden içeri aktarılabilir.

Tablo 150 - vRealize Orchestrator SSL Tasarım Karaları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-BYP-VRO-008 | Gömülü vRealize Orchestrator oluşumu vRealize Automation cihazının sertifikasını kullanacaktır. | vRealize Automation sertifikasının kullanımı yapılandırmayı kolaylaştırır. | Yoktur. |

vRealize Orchestrator bir veri tabanına ihtiyaç duyar. Bu tasarımda vRealize Automation cihazı içerisindeki gömülü PostgreSQL veri tabanı kullanılmıştır.

Tablo 151 - vRealize Orchestrator Veri Tabanı Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-BYP-VRO-009 | vRealize Orchestrator oluşumu vRealize Automation cihazı içindeki gömülü PostgreSQL veri tabanını kullanacaktır. | Gömülü PostgreSQL veri tabanının kullanılması performansta iyileşme ve tasarımın basitleşmesine izin verir. | Yoktur. |

Eklentiler sayesinde vRealize Orchestrator harici teknolojilere ve uygulamalara erişip onları kontrol edebilir. vRealize Orchestrator eklentisi ile harici bir teknolojiye erişerek onun içerisindeki nesne ve fonksiyonları iş akışlarına dahil etmek mümkün olur. Harici teknoloji eklentileri sanallaştırma yönetim araçları, eposta sistemleri, veri tabanları, kimlik hizmetleri ve uzak kontrol ara yüzleri olabilir. vCenter Server API ve eposta kabiliyetleri gibi teknolojileri vRealize Orchestrator bir set standart eklenti ile mevcut sisteme dahil edebilir.

Bunların yanı sıra açık eklenti mimarisi ile diğer uygulamalara erişim için özelleştirilmiş eklentiler de geliştirilebilir.

vCenter Server eklentisi ile birden çok vCenter Server oluşumu yönetilebilir. vCenter Server ortamındaki görevlerin otomasyonu için vCenter Server eklenti API’sini kullanan iş akışları oluşturulabilir. vCenter eklentisi vCenter Server operasyonlarını otomatikleştiren standart iş akışları kütüphanesi sağlar. Örnek vermek gerekirse sanal makinelerin oluşturulması, klonlanması, göç ettirilmesi ve silinmesi verilebilir.

Tablo 152 - vRealize Orchestrator vCenter Server Eklentisi Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-BYP-VRO-010 | vCenter Server’lar ile iletişimi kontrol edebilmek için vCenter Server eklentisi yapılandırılacaktır. | vCenter Server oluşumları ile iletişim için gerekli dolayısıyla iş akışları için de gereklidir. | Yoktur. |

* + - 1. vRealize Orchestrator Ölçeklenebilirliği

vRealize Orchestrator hem ölçek büyütmeyi hem de ölçek dağıtıklaştırmayı desteklemektedir.

Tek bir vRealize Orchestrator oluşumu 300 eş zamanlı iş akışı oluşumunu çalışır durumda olmasına müsaade eder. Bekleme ya da olay-bekleme durumundaki iş akışları bu rakama dahil değildirler. Uzun süre boyunca çalışması beklenen iş akışlarında iş akışı paletinden bekleme elemanları kullanılarak kaynaklardan tasarruf sağlanabilir. Bir vRealize Orchestrator oluşumu envanterinde 35.000 yönetilen sanal makine bulundurabilir. vRealize Automation cihazı sanal makinesinin işlemci ve hafızasını artırarak vRealize Orchestrator’ın ölçek büyütmesi sağlanabilir.

Mevcut tasarımda vRealize Orchestrator aynı ayarlar ile yapılandırılmış gömülü vRealize Orchestrator’a sahip vRealize cihazlarının kümelenmesi ile ölçek dağıtıklaştırma sağlanmaktadır. vRealize Orchestrator kümesi kullanmak eş zamanlı çalışan iş akışlarının sayısını artırırken yönetilen envanterdeki sanal nesnelerin sayısını artırmaz. vRealize Orchestrator sunucusu kümelenirken 5 aktif birime kadar sahip olan aktif-aktif küme seçilmelidir. VMware bu yapılandırmada en fazla 3 aktif birim olmasını tavsiye etmektedir.

Kümelenmiş bir vRealize Orchestrator ortamında diğer Orchestrator oluşumları çalışırken iş akışları değiştirilemez. Yeni bir iş akışı geliştirirken ya da değiştirirken vRealize Orchestrator istemcisi bağlanmadan önce diğer tüm Orchestrator oluşumları durdurulmalıdır.

Tablo 153 - vRealize Orchestrator Ölçek Dağıtıklaştırma Tasarım Kararları

|  |  |  |  |
| --- | --- | --- | --- |
| Karar ID | Tasarım Kararı | Tasarım Gerekçesi | Tasarım Sonuçları |
| SDDC-BYP-VRO-011 | vRealize Orchestrator aktif aktif küme olarak yapılandırılacaktır. vRealize Automation cihazı kümelendiğinde gömülü Orchestrator oluşumları otomatik olarak kümelenir. | Aktif aktif kümeler her iki vRealize Orchestrator sunucuların da iş akışı işletiminde eşit ve dengeli olarak yer almasını sağlar. | Yoktur. |

* 1. Operasyonlar Alt Yapısı Tasarımı
     1. vRealize Operations Manager Tasarımı
     2. vRealize Log Insight Tasarımı
     3. vSphere Data Protection Tasarımı
     4. Site Recovery Manager ve vSphere Replication Tasarımı
     5. vSphere Update Manager Tasarımı

1. Adams, Keith and Agesen, Ole. A Comparison of Software and Hardware Techniques for x86 Virtualization, 2006, [WWW]. Accessed on 8.12.2013. Available at: http://www.vmware.com/pdf/asplos235\_adams.pdf [↑](#footnote-ref-1)
2. blah blah blah [↑](#footnote-ref-2)
3. blah blah blah [↑](#footnote-ref-3)
4. [↑](#footnote-ref-4)
5. blah [↑](#footnote-ref-5)
6. blah [↑](#footnote-ref-6)
7. 6 ile aynı [↑](#footnote-ref-7)
8. blah [↑](#footnote-ref-8)
9. [↑](#footnote-ref-9)
10. [↑](#footnote-ref-10)
11. [↑](#footnote-ref-11)
12. 13 [↑](#footnote-ref-12)
13. [↑](#footnote-ref-13)
14. [↑](#footnote-ref-14)
15. [↑](#footnote-ref-15)
16. Kaltz C [↑](#footnote-ref-16)
17. [↑](#footnote-ref-17)
18. 21 [↑](#footnote-ref-18)
19. VMware Compatibility Guide, 2017 [↑](#footnote-ref-19)