

T.C.
BEYKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ BİLİM DALI

**BULUT BİLİŞİM'DE SUNUCU DEVRE DIŞI
KALDIĞINDA VERİ KAYBINI ÖNLEYEN YÖNTEM**
Yüksek Lisans Tezi

Tezi Hazırlayan:
Selim AKSOY

İSTANBUL, 2014

T.C.
BEYKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ BİLİM DALI

**BULUT BİLİŞİM'DE SUNUCU DEVRE DIŞI
KALDIĞINDA VERİ KAYBINI ÖNLEYEN YÖNTEM**
Yüksek Lisans Tezi

Tezi Hazırlayan:
Selim AKSOY
Öğrenci No:
110820010

Danışman:
Yrd.Doç.Dr. Ediz ŞAYKOL

İSTANBUL, 2014

YEMİN METNİ

Yüksek lisans tezi olarak sunduğum “Bulut Bilişim’de Sunucu Devre Dışı Kaldığında Veri Kaybını Önleyen Yöntem” başlıklı çalışmanın bilimsel ahlak ve geleneklere uygun şekilde tarafımdan yazıldığını, yararlandığım eserlerin tamamının kaynaklarda gösterildiğini ve çalışmamın içinde kullanıldıkları her yerde bunlara atıf yapıldığını belirtir ve bunu onurumla doğrularım. 15/02/2014

Aday: Selim AKSOY

T.C.
BEYKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZ SAVUNMA SINAVI SONUÇ TUTANAĞI

Beykent Üniversitesi
Fen Bilimleri Enstitüsü Müdürlüğü'ne,

Aşağıda tez adı belirtilen yüksek lisans öğrencisi. 110820010....no'lu Selim..Aksoy'in 10/03/2014 tarihinde yapılan tez savunma sınavı¹ sonucunda. 50. dakika süreyle sunduğu ve savunduğu tezi hakkında² oybirliğiyle, KABUL kararı verilmiştir.

Bilgilerinize saygılarımızla arz ederiz.

Anabilim Dalı : Bilgisayar Mühendisliği
Programı : Bilgisayar Mühendisliği
Tez Başlığı³ : Bulut Bilişim ile Sunucu Devre Dışı Kaldığında Veri Kaybını Önleyen Yöntem

Tez Sınav Jürisi

Öğretim Üyesi

İmza

Danışman : Yrd. Doc. Dr. Ediz Saykol
Üye : Doc. Dr. Gökhan Silahterioğlu
Üye : Yrd. Doc. Dr. Turhan Karagüler

¹ Jüri üyeleri söz konusu tezin kendilerine teslim edildiği tarihten itibaren en geç bir ay içinde toplanarak öğrenciyi tez savunma sınavına alır. Belirlenen günde yapılamayan jüri toplantısı, katılanların hazırladığı bir tutanakla enstitü yönetimine bildirilir. Bu durumda jüri en geç onbeş gün içinde toplanarak adayı tez savunma sınavına alır. Tez savunma sınav süresi en az 45 dakikadır. Yüksek lisans tez savunma sınavı, tez çalışmasının sunulması ve bunu izleyen soru-yanıt bölümlerinden oluşur ve dinleyiciye açıktır. (Beykent Lisansüstü eğitim ve Öğretim Yönetmeliği-Madde30-3)

² Tez sınavının tamamlanmasından sonra jüri, tez hakkında "kabul", "düzeltme" veya "red" kararı verir. Jüri başkanı, jüri üyelerince imzalanmış sınav tutanağını, tez sınavını izleyen üç gün içinde ilgili enstitü yönetimine teslim eder. Tezi başarısız bulunan öğrencinin Enstitü ile ilişkisi kesilir. Tezi hakkında düzeltme kararı verilen öğrenci en geç üç ay içinde gerekli düzeltmeleri yaparak ve yönetmelikte belirtilen usullere uygun olarak tezini aynı jüri önünde yeniden savunur. Bu savunma sınavında da tezi kabul edilmeyen öğrencinin enstitü ile ilişkisi kesilir. (Beykent Lisansüstü eğitim ve Öğretim Yönetmeliği-Madde30-4)

³ İleride doğabilecek aksaklıkların engellenmesi için tezin başlığının yazılması gerekmektedir.

Adı ve Soyadı : Selim Aksoy
Danışmanı : Yrd. Doç. Dr. Ediz ŞAYKOL
Türü ve Tarihi : Yüksek Lisans / Tez, 2014
Alanı : Bilgisayar Mühendisliği
Anahtar Kelimeler : Bulut Bilişim, Yüksek Erişilebilirlik, Servis Olarak Altyapı

Özet

BULUT BİLİŞİM'DE SUNUCU DEVRE DIŞI KALDIĞINDA VERİ KAYBINI ÖNLEYEN YÖNTEM

Bu tez çalışmasında bulut altyapısı oluşturulup, fiziksel sunucuda herhangi bir sorun olduğunda bu fiziksel sunucu üzerindeki sanal sunucuların etkilenmesini asgari düzeye indirecek bir sistem oluşturmak hedeflenmiştir. Ayrıca depolama tarafında kesinti yaşanmaması için kullanılan yöntemleri de ekleyerek daha çevik ve kesintisiz bir yapı oluşturulmuştur.

Name and Surname : Selim Aksoy
Supervisor : Yrd. Doç. Dr. Ediz ŞAYKOL
Degree and Date : Master, 2014
Major : Computer Engineering
Key Words : Cloud Computing, High Availability, Infrastructure As A Service

Abstract

AN IMPLEMENTATION TO PREVENT DATA LOSS ON PHYSICAL SERVER FAILURE IN THE CLOUD

In this thesis, I created a cloud architecture and when a problem occurs in the physical server, the virtual servers on this physical server will continue to run with minimum down-time. In addition, a method used to prevent the storage failure, so that will provide high-availability system architecture.

İÇİNDEKİLER

Sayfa No.

ÖZET	i
ABSTRACT	ii
ŞEKİLLER LİSTESİ	v
KISALTMALAR	viii
1.GİRİŞ	1
2.BULUT BİLİŞİM	2
2.1. Bulut Bilişim Nedir?	2
2.2. Bulut Bilişim Türleri	3
2.2.1. Genel Bulut	3
2.2.2. Özel Bulut	3
2.2.3. Hibrit Bulut	3
2.3. Bulut Bilişim Modelleri.....	4
2.3.1. Infrastructure As A Service (IAAS) - Servis Olarak Altyapı	4
2.3.2. Platform As A Service (PAAS) - Servis Olarak Platform	4
2.3.3. Software As A Service (SAAS) - Servis Olarak Yazılım.....	5
2.4. Bulut Bilişim Sağlayıcıları	5
2.4.1. Amazon EC2 (Elastic Compute Cloud).....	5
2.4.2. Microsoft Azure IAAS.....	6
2.4.3. Rackspace	7
2.4.4. HP Genel Bulut	7
2.4.5. Cloudturk	8
2.5.Bulut Bilişim Kullanım Senaryoları.....	9
2.5.1. Bireysel Kullanım	9
2.5.2. Kurumsal Kullanım.....	9
2.5.3. Tekil Sağlayıcı Bulut Servisi	10
2.5.4. Çoğul Sağlayıcı Bulut Servisi.....	10
2.5.5. Servis Geliştirici.....	11

	Sayfa No.
3.GELİŞTİRİLEN YÖNTEM.....	11
3.1. Sunucu	11
3.2. Problemin Etkisi	11
3.3. Sunucu Sanallaştırma	13
3.4. Kümelenmiş (Clustered) Yapı	15
3.5. Veri Replikasyonu	16
4.PERFORMANS DEĞERLENDİRMESİ	18
4.1. Etkin Ve Verimlilik	18
4.1.1. Bulut Kurulumu	19
4.1.1.1. Mimari Yapı Hakkında	19
4.1.1.2. Kurulum İçin Seçilen Yazılım Ve Versiyonları.....	20
4.1.1.3. Donanım Gereksinimleri.....	20
4.1.1.4. Ağ Yapılandırması	20
4.1.1.5. Bulut Kurulum Adımları.....	21
4.1.2. Yüksek Erişilebilir Depolama Kurulumu	34
4.1.3. Canlı Taşıma Yapılandırması	37
4.1.3.1. Libvirt ve Nova Yapılandırması	37
4.1.4. Sistemin Test Edilmesi	41
5.SONUÇ.....	43
KAYNAKLAR	44

ŞEKİLLER LİSTESİ

Sayfa No.

Şekil.1. Bulut bilişim genel.....	2
Şekil.2. Bulut bilişim türleri.....	4
Şekil.3. Bulut bilişim modelleri	5
Şekil.4. Lokal & EBS karşılaştırması	6
Şekil.5. Amazon EC2 servis kesintisi	12
Şekil.6. Sanallaştırma görünümü	13
Şekil.7. Veri merkezinde barındırılan sunucular	14
Şekil.8. Genel depolama yapısı.....	16
Şekil.9. Veri replikasyonu senkronize çalışma mantığı.....	17
Şekil.10. Depolama karşılaştırması	18
Şekil.11. RabbitMQ yükleme işlemi.....	21
Şekil.12. MySQL yükleme işlemi.....	21
Şekil.13. MySQL dışarıdan erişim sağlanması.....	21
Şekil.14. MySQL servisinin yeniden başlatılması.....	21
Şekil.15. Nova servisinin kurulumu	22
Şekil.16. Nova bileşenlerinin kurulumu	22
Şekil.17. Nova depolama servisi kurulumu	22
Şekil.18. Nova sanallaştırma servisi kurulumu.....	22
Şekil.19. Nova lxc sanallaştırma servisi kurulumu.....	23
Şekil.20. Nova konfigürasyon dosyası düzenlemesi	23
Şekil.21. Nova sanallaştırma türü değiştirilmesi	23
Şekil.22. Nova sanallaştırma servisi yeniden başlatılması	23
Şekil.23. Nova servisinin devredışı bırakılması	23
Şekil.24. Glance veri tabanı işlemleri	24
Şekil.25. Glance modülü kurulumu	24
Şekil.26. Keystone veritabanı işlemleri	24
Şekil.27. Keystone modülü kurulumu	24

Şekil.28. Node2 sanallaştırma servisi kurulumu.....	24
Şekil.29. Node3 sanallaştırma servisi kurulumu.....	24
Şekil.30. Host dosyası düzenleme komutu	25
Şekil.31. Host dosyası konfigürasyonu.....	25
Şekil.32. Ağ arayüzü düzenleme işlemi.....	25
Şekil.33. Node1 ağ arayüzü ayarları	25
Şekil.34. Node2 ağ arayüzü düzenleme işlemi	26
Şekil.35. Node2 ağ arayüzü ayarları	26
Şekil.36. Nova konfigürasyon dosyası komutu	26
Şekil.37. Nova konfigürasyon dosyası	27
Şekil.38. Glance kayıt dosyası düzenleme.....	28
Şekil.39. Glance kayıt dosyası konfigürasyonu	28
Şekil.40. Keystone konfigürasyon dosyası	29
Şekil.41. Keystone konfigürasyon dosyası yapılandırması	29
Şekil.42. Glance ve Nova konfigürasyon dosyaları.....	29
Şekil.43. Sunucuyu yeniden başlatma işlemi.....	29
Şekil.44. Nova veri tabanı tabloları oluşturma işlemi.....	30
Şekil.45. Glance ve keystone veri tabanı tabloları oluşturma işlemi.....	30
Şekil.46. Dosya kopyalama işlemi.....	30
Şekil.47. Nova konfigürasyon dosyası komutu	30
Şekil.48. Nova konfigürasyon dosyası düzenleme işlemi	30
Şekil.49. Servislerin kontrol edilmesi.....	31
Şekil.50. Konfigürasyon ayarlarının listelenmesi.....	31
Şekil.51. Sanal ağ oluşturulması	31
Şekil.52. Dışa açık ip oluşturulması	32
Şekil.53. Depolama servisi yeniden başlatma işlemi.....	32
Şekil.54. Kontrol paneli yükleme işlemi	32
Şekil.55. Önbellekleme yükleme işlemi	33
Şekil.56. Ubuntu tema yüklemesi	33

Şekil.57. Apache servisi yeniden başlatma işlemi	33
Şekil.58. Kontrol paneli giriş ekranı	33
Şekil.59. Sunucuların depolamaya eklenmesi	34
Şekil.60. Sunucuların depolamadaki durumu	34
Şekil.61. Depolama oluşturulması ve kopya sayısı	35
Şekil.62. Depolamanın başlatılması.....	35
Şekil.63. Depolama bilgilerinin kontrol edilmesi	35
Şekil.64. Sunucu disk kullanım durumu	36
Şekil.65. Depolama test işlemi	36
Şekil.66. Depolama test sonucu	36
Şekil.67. Libvirt konfigürasyonu düzenleme komutu	37
Şekil.68. Libvirt-bin konfigürasyonu düzenleme komutu	37
Şekil.69. Diğer libvirt-bin konfigürasyonu düzenleme komutu	37
Şekil.70. Libvirt servisi yeniden başlatma komutu.....	38
Şekil.71. Sistemin kontrolü.....	38
Şekil.72. Aktif olmayan bulut sisteminde sunucu kurtarma işlemi	39
Şekil.73. Bulut sistemi ve depolama çalışma şeması.....	40
Şekil.74. Nova canlı taşıma satırı	40
Şekil.75. Sanal sunucuları listeleme komutu	41
Şekil.76. Sistemdeki sanal sunucular.....	41
Şekil.77. Sanal sunucunun detaylarını görüntüleme komutu	41
Şekil.78. Sanal sunucu detayları	42
Şekil.79. Canlı taşıma komutu.....	42
Şekil.80. Canlı taşıma işlemi sonucu	42

KISALTMALAR

BT	: Bilişim Teknolojileri
RPO	: Recovery Point Objective
RTO	: Recovery Time Objective
RPM	: Revolutions Per Minute
API	: Application Programming Interface
IaaS	: Infrastructure as a Service
PaaS	: Platform as a Service
SaaS	: Software as a Service
VLAN	: Virtual Local Area Network

1. GİRİŞ

Bulut bilişim, ortak kullanılan kaynaklar üzerinde, düşük maliyet ve servis sağlayıcı etkileşimi ile, hızlıca kontrol edilebilir ve kolay yönetilebilir bir hizmettir. Ayarlanabilir bilişim kaynaklarının paylaşılır havuzuna, istendiğinde ve uygun bir şekilde ağ erişimi sağlayan, son kullanıcının bakım gibi konularla uğraşmak zorunda kalmayıp esas uzmanlık alanlarına odaklanabilmelerini sağlayan bir modeldir.

Bulut bilişim bir yandan karmaşık bilgi teknolojileri ortamlarını basitleştirip farklı uygulama ve sistemlerin birlikte uyum içerisinde çalışmasını sağlarken, öte yandan girdi ve gider tasarrufu sağlamaktadır. Böylece yatırımların geri dönüşü hızlanmakta, kaynaklar daha verimli kullanılmakta, firma ve bireylerin rekabet gücü artmaktadır. Kullandıkça öde modeli ile yeni kurulmuş bir işletmenin bilgi teknolojileri harcamalarını sınırlı bir düzeyde tutmayı sağlar. Bulut sayesinde uygulamaların güncelleme ve bakımı kolaylaşmakta, arıza giderme süreleri kısalmakta, ürünlerin kullanım ömrü uzamakta, enerji tüketimi azalmakta, firmalar bilişim kapasitelerini basit bir biçimde ve süratle artırabilmektedir. Seçmiş olduğunuz sistemi bir kaç dakikada ayağa kaldırıp test işlemini gerçekleştirebilmekte, bu sayede zamandan ve maliyetten tasarruf etmiş oluyorsunuz.

Yazılımsal veya donanımsal hata nedeniyle sunucuların arızalanması durumunda kesinti yaşanmaması ve işlemlerinin mümkün olduğunca hızlı bir şekilde sorunsuz çalışması çok önemlidir. Kullanılan depolama yerlerinin birleştirilerek tek bir depolama alanı haline getirilip, herhangi bir felaket senaryosunda depolama tarafında kesinti (downtime) yaşanmaması için kullanılan yöntemlerini de ekleyerek daha çevik ve kesintisiz bir yapı oluşturulur. Oluşturulan sistem her durumda çalışmak üzere dizayn edilmiştir. Oluşturulan sanal sunucuların birden fazla fiziksel sunucu üzerine yayarak, herhangi bir fiziksel sunucu üzerinde oluşabilecek hatadan dolayı, çalışan sistemin etkilenmemesi için çözümler aranmaktadır. Böyle durumlarda yedekli olmayan mimarilerin yarı yolda kalma ihtimali çok fazladır.

2. BULUT BİLİŞİM

2.1. Bulut Bilişim Nedir?

Bulut teknolojisi en yalın haliyle internet üzerinden erişimde bulunulan yazılım uygulamaları, veri depolama hizmeti ve işlem kapasitesidir. Bu teknolojiye şirketlerin bilişim alanında kullanılan yeteneklere ne zaman isterse ya da ihtiyaç duyarsa o zaman erişmesi esas alınır. Bilgisayar gücünün elektriğin satın alınır gibi veya suyun kullanılması gibi kullanılmasıdır. Özellikle Türkiye’de bulut bilişim kavramı çokça kullanılmaya başlandı. Ülkemizde telekomünikasyonfirmalarının adları bulut bilişim ile daha sık anılır oldu. Son birkaç yıl içerisinde, bulut bilişim, bilişim dünyasında her alanda hızla yaygınlaşmakta ve kullanılmaktadır. Bulut bilişim, kişi ve kurumlara merkez serbestliği sağlar ve üstelik sabit bedeller yerine kullanılan kapasite ve hizmet kadar ödeme mantığı ile çok daha ekonomik hale gelir. Genel olarak bu yapılar son kullanıcı sanal makinesini yalnızca çalıştırdığı zamanlar ücret almayı ve sadece kullandığınız kadar ödemenizi sağlayacak şekilde tasarlanmıştır.

Veri merkezlerinde sanallaştırmanın uygulanması, kaynakların daha verimli bir şekilde kullanılmasını aynı zamanda maliyetlerin de düşmesini sağlamaktadır. Bu düşen maliyetler son kullanıcıya da yansıtılmaktadır.



Şekil.1. Bulut bilişim genel. [1]

2.2. Bulut Bilişim Türleri

2.2.1. Genel Bulut

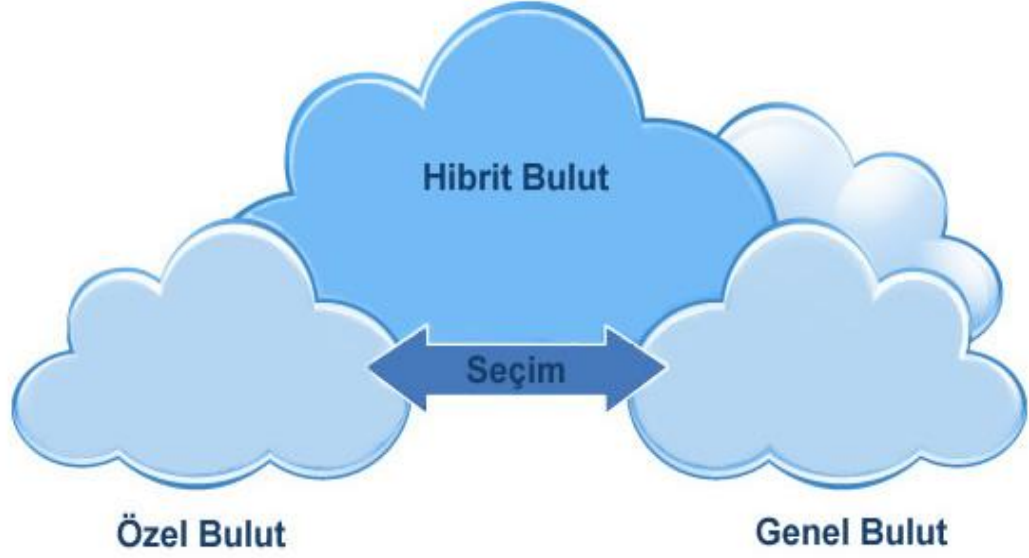
Son kullanıcının ihtiyacı olan kaynağı istediği gibi belirleyip internet üzerindeki sunucuları kullanabilmesidir. Genel bulut çözümünde depolama ve diğer kaynaklar bir bulut hizmet sağlayıcısı tarafından kullanıcılara sunulur. Kontrol paneli, API ve mobil uygulamalar ile sanal sunucular oluşturulup, yönetilebilir.

2.2.2. Özel Bulut

Şirket bünyesinde oluşturulmuş sunucular ile verilen bulut hizmetidir. Şirketler kendi bulutunu kurar ve dışarıya kapalı olan bulut, şirket içinde ortak olarak kullanılır. Bu sayede güvenlik sistemleri üzerinde daha fazla denetim ve müdahale imkanı olduğu için şirketin veri güvenliği konusundaki kaygıları minimize edilir.

2.2.3. Hibrit Bulut

Hibrit bulut iki veya daha fazla bulutun birleşimidir. Bir şirketin verilerin güvenliği gibi nedenlerden dolayı tüm ortamını genel buluta taşımayıp, özel bulut ile birlikte kullanması durumunda ortaya çıkan yapıdır. Bu farklı yapıdaki bulutlar ayrı bulunmaktadır ancak birbirlerine bağlıdırlar. Hibrit bulutlarda kurum içi uygulamalar esnek, güvenli ve belirli olmalıdır.[2] Hibrit bulut sağlayıcıları bu ihtiyacı bulut tabanlı hizmetlerin hata toleransı ve ölçeklendirilebilirliği ile sağlar.



Şekil.2. Bulut bilişim türleri.

2.3. Bulut Bilişim Modelleri

2.3.1. Infrastructure As A Service(IAAS)-Servis Olarak Altyapı

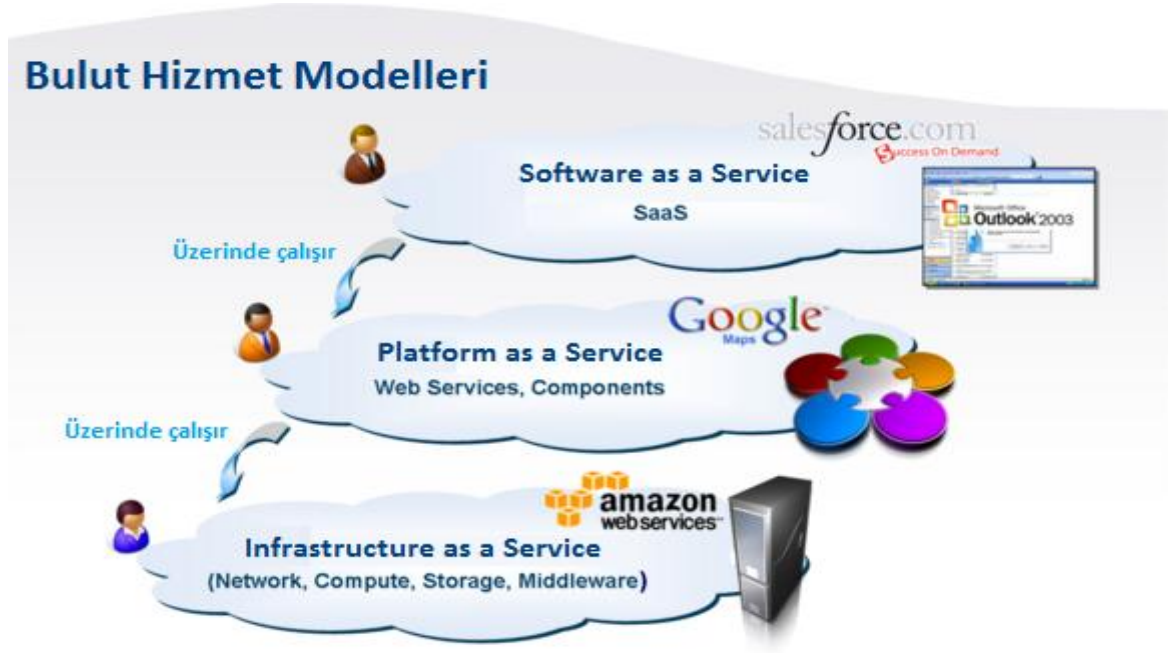
En temel bulut hizmet modelinde, bulut sağlayıcıları, sunucuları, fiziksel veya sanal makineler olarak sunarlar. İşlemci gücü, veri saklama ve ağ kaynaklarının servis olarak sunulduğu modeldir. Amazon EC2, Rackspace, Turkcell Super Bulut, TurkTelekom Bulutt, HP Cloud, Cloudturk bu modele örnektir.

2.3.2. Platform As A Service(PAAS)-Servis Olarak Platform

Müşteriye ihtiyaçları doğrultusunda hazırlanmış bir sunucunun ve bu sunucu üzerinde kurulmuş bir uygulamanın internet yoluyla sunulmasıdır. Uygulama geliştirme ve test işlemleri için uygundur. Kullanıcı platformun çalıştığı altyapıyla uğraşmak yerine hazır bir platforma sahip olarak hemen kullanmaya başlayabilir. Arızalanma, güncelleme gibi risk ve maliyetlerde ortadan kalkmış olur. Google App Engine, Salesforce, Jelastic, Microsoft Azure bu modele örnektir.

2.3.3. Software As A Service(SAAS)-Servis Olarak Yazılım

Bulut sağlayıcıları bulutta uygulama yazılımını yüklemeyi ve işletmeyi yapar, bulut kullanıcıları bulut hizmeti alanlardan yazılıma erişirler. Bulut kullanıcıları uygulamanın üzerinde çalıştığı bulut altyapısı ve platformunu yönetmezler.[3] Exchange hosting, Gmail, Atlassian bu modele örnektir.



Şekil.3. Bulut bilişim modelleri.[4]

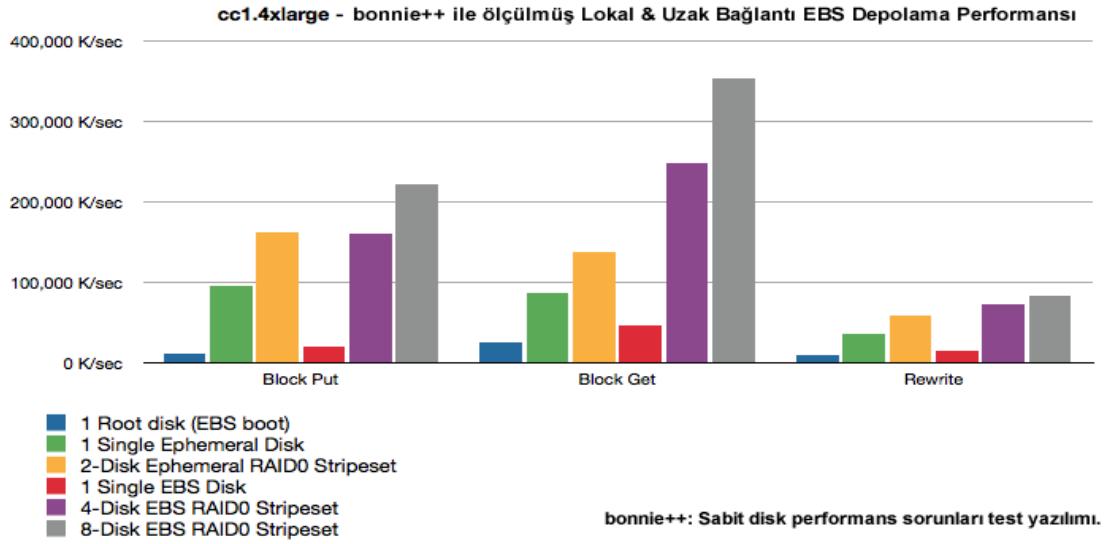
2.4. Bulut Bilişim Sağlayıcıları

2.4.1. Amazon EC2(Elastic Compute Cloud)

Amazon EC2, dünya elektronik ticaret devi Amazon firmasının bulut bilişim ürünlerinin en üstünde bulunan bir teknolojidir. Amazon EC2 IaaS modeliyle ve kendi veri merkezlerinde barındırdığı sunucuları ile farklı büyüklükteki firmalara bulut hizmetini vermektedir. Sistem neredeyse sınırsız güce sahip bir bilgisayar işlevi görür. Bu sistemde çalışan bir hizmet, hesaplama gücünün küçük bir kısmını kullanabileceği

gibi, talep edilirse otomatik olarak hesaplama gücünü genişletir ve büyük miktarlarda işlem yapabilir. Kullanıcıya tek bir bilgisayar gibi görünse de bu yük, internet üzerinde birbirleriyle bağlantılı birçok sunucuya dağıtılır. Amazon EC2 Micro, Small, Medium, Large ve Extra Large Instance olarak isimlendirilmiş farklı donanım ortamı sunmaktadır.

Amazon EC2 bulut hizmetinde iki türlü depolama kullanılmaktadır. Lokal depolama seçilerek oluşturduğunuz sanal sunucu ilgili fiziki sunucunun diskinde saklanır. Dolayısıyla bu sunucuda meydana gelebilecek sorunlarda sanal makineniz etkilenecek belki de çalışmayacaktır. Diğer depolama seçeneği ise EBS'dir. EBS elastic block storage kelimelerinin kısaltılmış şeklidir. EBS depolama verinin birden fazla kopyasını alır, böylece veri kaybının önüne geçilmiş olur. Bu depolama sisteminde donanımbaşarısız olsa dahi, ilgili depolama birimi erişilebilir kalır.



Şekil.4. Lokal & EBS karşılaştırması.

2.4.2. Microsoft Azure IAAS

Windows Azure IaaS, Microsoft'un Genel Bulut uygulama ve altyapı platformudur.

Windows Azure, ihtiyaca göre hesaplama veya depolama kabiliyetleri, web servislerinin ölçeklenmesi, yönetilmesi ve birleşik uygulamalar sağlayarak uygulamaların bakım ve operasyonunu basitleştiriyor.[5] Windows Azure IaaS ile Windows ya da Linux sanal makineler oluşturabiliriz. Bu sanal makineleri istersek Azure galerisinden veya istersek kendi oluşturduğumuz imajları yükleyip bu imajlar sayesinde sanal makineler oluşturabiliriz. Sanal ağlar ve diskler oluşturabilir veya kaldırabiliriz. Windows Azure, uygulamalarınızı herhangi bir boyuta kolayca ölçekleyebilmenizi sağlar. Bu, kaynakları birkaç dakika içerisinde sağlamanıza izin veren, tam otomatik bir platformdur. Kaynak kullanımınızı, ihtiyaçlarınıza bağlı olarak esnek bir şekilde büyütebilir ya da küçültebilirsiniz. Sadece uygulamanızın kullandığı kaynaklar için ödeme yaparsınız. Kullanıcı altyapının alımı ve bakımı için harcama yapmasına gerek kalmaz. Windows Azure dünyada birden fazla veri merkezinde mevcuttur ve sanal sunucularınızı en uygun veri merkezinde konumlandırabilmenize imkan sağlar. Windows Azure üzerinde şu anda sunucu yedekleme, sunucuya kontrol panelindeki konsol üzerinden bağlanmak ve ağ ekleme özellikleri gibi eksikleri vardır.

2.4.3. Rackspace

Rackspace'in OpenStack platformu, çatısı altında bulundurduğu bir çok servisi tek bir kontrol panel üzerinden sunarak, bir çok ihtiyacınıza aynı ekran üzerinden çözüm sunmaktadır. Sanal sunucular için yedekleme, DNS servisi, yük dengeleyici ve otomatik ölçeklendirme gibi bir çok hizmeti merkezi arabirime ekleyebilmiş konumdalar. Rackspace aynı sunucu üzerinde kaynak azaltımı yapmanıza izin vermemekte. Sunucunuzun işlemci, bellek ya da disk kaynağını arttırmak isterseniz de tüm kaynakları bir üst paket özelliklerinde arttırmanız gerekiyor.

2.4.4. HP Genel Bulut

HP Genel Bulut ile sanal sunucular üzerinden isteğe bağlı olarak BT hizmetleri alınabiliyor, ihtiyaca göre çevrimiçi depolama alanı sağlayıp ve son kullanıcılara hızlı

önbellek erişimi teknolojisi ile buluttan hızlı veri indirme olanağı tanıyor. Böylece kullanıcılar BT hizmetlerini birkaç dakikada yapılandırarak yalnızca kullandıkları kadar ödeme modelinin kullanımına olanak sağlıyor.

HP Genel Bulut OpenStack teknolojisinden yararlanarak iş yükündeki anlık artış taleplerini “bursting” teknolojisi ile sağlamanın yanı sıra KVM kaynak havuzlarını da destekliyor. Linux ve Windows şablon imajları kullanılarak istenilen sanal sunucu kolayca oluşturulabiliyor.

Turkcell Süper Bulut hizmetinin HP Bulut Çözümleri ile oluşturulduğu bilinmektedir. Turkcell işletmelere Turkcell bünyesindeki sunucular, depolama birimleri ve ağ sistemleri üzerinden sunulan en yeni BT teknolojilerine isteğe bağlı hizmet modeliyle erişme imkanı sunacak. Kurumsal firmalar ve KOBİ’ler, Turkcell’den IaaS hizmeti alabiliyor veya sadece kullanım trafiğinin arttığı saatlerde kendi ofislerindeki veri merkezlerini Turkcell’den kiralanan IaaS sistemleriyle takviye edebiliyor. Bulut bilişim üzerinden hizmet olarak altyapı çözümü Turkcell’in veri merkezleri tarafından sağlanıyor ve yönetiliyor.

Türk Telekom Grubu’nun bulut hizmet altyapısı olarak HP Bulut Çözümleri’ni kullandığı bilinmektedir.

2.4.5. Cloudturk

Cloudturk, Ocak 2012 Türkiye’de sunucu altyapı hizmet sağlayıcılığı yatırımını gerçekleştirdiğini duyurdu. Cloudturk 72 sunucu ile bu hizmeti vermeye başladığı ve Maltepe’de Anadolu Bilişim Veri Merkezi’nde sunucularını barındırdığı bilinmektedir.

Cloudturk’ün sunduğu teknoloji küçük, orta ve büyük işletmelerin ciddi avantajlar kazanmasına olanak tanıyor. Şirketler sunucu ihtiyaçları olduğunda yeni sunucu almak yerine Cloudturk altyapısını kullanmayı tercih edebiliyorlar. Büyük

ölçekli şirketler ise koşullarına göre belirli parçalarını buluta taşıırken, bazen de tüm altyapılarını Cloudturk sunucu altyapısı üzerinde yarattıkları özel buluta taşıma şansına sahipler. Ek olarak, Cloudturk, şirketlere mali açıdan da büyük yararlar getiriyor. Her işlem için harcanan kaynak ayrı ayrı hesaplandığından, kullanılmayan işlem kaynakları faturalandırılmıyor.

2.5. Bulut Bilişim Kullanım Senaryoları

2.5.1. Bireysel Kullanım

Bireysel kullanımda kullanıcı bulut hizmetlerinden faydalanır. Arka planda servislerin nasıl çalıştığıyla ilgilenmez. Dünyadaki veri paylaşımının arttığı ve bununla birlikte arşivleme ihtiyacının da paralel olarak artacağı düşünülürse bireysel alanda da bulut bilişim kullanımının artacağı tahmininde bulunmak yanlış olmayacaktır. Güncel yaşantıda sıkça kullanılan ve kullandıkça öde kolaylığıyla en yaygın kullanılan iş modelidir. “Gmail” ve “iCloud” bu senaryoya örnekgösterilebilir. iCloud, dünya çapında 200 milyonun üzerinde iPhone ve iPad cihazı satan Apple’ın bireysel bulut servsidir. Kullanıcıların müziklerini, fotoğraflarını, uygulamalarını, takvimlerini ve belgelerini Apple sunucularında tutarak Apple bilgisayarlar dahil tüm Apple cihazları arasında senkronize edilmesini sağlıyor. Örneğin kullanıcı iPhone cihazına bir müzik dosyası indirdiğinde veya iPad’inde bir metin belgesi oluşturduğunda bu dosya tüm Apple cihazları arasında kablosuz olarak senkronize ediliyor.

2.5.2. Kurumsal Kullanım

Bu senaryo içerisinde birden çok alt senaryo barındırmaktadır; Bunların ilkinde, işletme son kullanıcılara kullanırmak üzere bulut servislerinden faydalanır. Elektronik posta ya da veri paylaşımı servisleri örnek olarak verilebilir. İkinci senaryoda, BT fonksiyonlarının bazılarının bulut servislerine aktarıldığını görürüz. Veri saklama, yedekleme ve veri tabanı servisleri ilk anda verebilen örneklerdir. Kurum BT servislerini

birbirinden bağımsız hale getirebildiği ölçüde bulut servislerinden yararlanma oranı artacaktır. Üçüncü senaryoda, kurum BT altyapısını tamamen bulut üzerine aktarılır. Bir ve ikinci senaryolar dışında kalan her şey bulut üzerindeki sanal sunuculara aktarılır. Dördüncü senaryoda ise kurum verimliliğini artırmak maliyetlerini düşürmek için servis sağlayıcı mimarisini kendi veri merkezinde hayata geçirir. Bir ve ikinci senaryolar yine bu senaryo ile kombine edilebilir. Kurumların bulut servislerinden hangi senaryoları kullanabilecekleri, hangi oranda yararlanabilecekleri, ne gibi ihtiyaçları olduğunun analizini yaptırmaları gerekmektedir.[6] Ayrıca kurumun stratejisi ve kurum BT altyapısının esnekliği önemlidir. BT altyapı esnekliğini sağlamak açısından bir tür servis odaklı mimari modeli oluşturulabilir.

2.5.3. Tekil Sağlayıcı Bulut Servisi

Servis sağlayıcı tüm BT altyapısını, yani altyapı, uygulama platformu ve yazılım servislerini tek elden sunar. Sanallaştırma ve özel bulut söz konusu olduğunda hizmetleri tek bir sağlayıcıdan almak kuruma avantajlar sağlayabilir. Örneğin resmi prosedürlerin tamamlanması daha az süre alıp, maliyeti bakımından daha uygun olacaktır. Tekil sağlayıcı bulut servisi ile çalışanlarınıza daha az eğitim gerektirir ve kendi çalışmalarınıza daha iyi odaklanabilirsiniz. Donanımsal olarak çıkacak problemler ile zaman kaybetmezsiniz.

2.5.4. Çoğul Sağlayıcı Bulut Servisi

Servis sağlayıcı servisini bir ya da daha çok servis sağlayıcının sunduğu servisler üzerine kurar. Örneğin; fotoğraf paylaşım servisi olan Instagram veri saklama hizmetini Amazon S3'den almaktadır. Ama bu servisi kullanan tüketici yalnızca Instagram ile muhatap olur, gerisinden haberi olmaz.

2.5.5. Servis Geliştirici

Servis geliştirici (kurum ya da birey) servis sağlayıcı tarafından sağlanan geliştirme ortamı ile belli servis sağlayıcı çalıştırma ortamı için uygulamasını ya da servisi geliştirir. Bu uygulama/servis arka planda birden çok servis sağlayıcının sunduğu servis ya da altyapıyı kullanabilir. Örneğin; Google Apps Engine için geliştirme yapan bir geliştiriciyi düşünelim. Bu geliştiricinin büyük miktarda saklama alanına ve ilişkisel veri tabanına ihtiyacı olduğunu varsayalım. Google Apps Engine bu gereksinimleri karşılayamaz ama Amazon S3 ve EC2 karşılayabilir. Sonuçta Google Apps Engine üzerinde çalışan uygulama/servis aynı zamanda Amazon servislerini de kullanıyor olur.

3. GELİŞTİRİLEN YÖNTEM

3.1. Sunucu (Node)

Bulut bilişim hizmetlerini hayata geçirmek için gerekli işlem, depolama ve iletişim gücümü sağlamakla görevli birimlerdir.

Sistemin kurulu olduğu, fiziksel sunucu olarak varsayacağımız makinelerdir. Sanal makineler bu sistem üzerinde oluşturulacaktır. Standart sistemlerinin kurulumunda buna bulut sistemleri de dahil fiziksel sunucular yedekli şekilde değildir. Dolayısıyla herhangi bir fiziksel makine bozulduğunda veya bakıma alındığında üzerindeki sanal makineler çalışamaz durumda olur. Dolayısıyla kesinti (downtime) meydana gelir.

3.2. Problemin Etkisi

Problemin etkisi firmanın büyüklüğüne göre değişmekle birlikte, e-ticaret gibi üzerinde yüzlerce alış-veriş işlemin yapıldığı yerlerde veya hastahane otomasyonu gibi veritabanının bulut sunucularda saklandığı durumlarda bu veritabanına bağlı tüm sistemlerde kesinti oluşur. Tüm veriler büyük bir diskte saklanamaz. Eğer herhangi bir

şekilde bu diskte sorun oluşursa tüm verilerin kaybolması riski vardır. Yedekli olmayan mimarilerde yarıyolda kalma ihtimali çok fazladır.

Amazon EC2 hizmetlerini kullanan Türkiye dahil olmak üzere dünyanın birçok yerinde kullanıcılar ağırlıklı olarak Twitter üzerinden Instagram, Pinterest gibi servislerde kesinti raporlamaya başladılar. Sonradan anlaşıldı ki bu servislerin yanında Netflix ve Heroku gibi servisler de dahil olmuş. Tüm bu servislerin ortak noktası Amazon tarafından sunulan bulut servisleri üzerinde barındırılmaları. Eğer bu şirketler yük dengeleme hizmetini kullanmıyorsa ve tüm verilerini bir merkezde tutuyorsa, o merkez zarar gördüğünde uzun süre kullanım dışı kalması mümkün.

Amazon ve benzeri bulut servisleri bu tür kesintileri her zaman yaşıyor, ama oluşan kesintiler yük dengeleme sistemleriyle bir şekilde en aza indiriliyor. Yani bir diğer servisin boştaki kalan sunucusu, üzerinde yoğun istek olan sunucuya bir süre için destek veriyor ve olay çözülüyor. Ancak kesinti sırasında aşırı miktarda sunucunun devre dışı kalması, yük dengeleme sisteminin de düzgün çalışmaması nedeniyle sistemler genel bir kaosa sürüklenmesine yol açıyor. Bu tür sistemlerin her zaman fiziksel sunucu bakımından hataya karşı direnç göstereceği maksimum makine sayısı vardır. Örneğin sunucular aralarından ikili olacak şekilde kümelenendirilmişler ise aynı kümedeki bir sunucunun devre dışı kalması sistemin sürekliliğini etkilemeyecektir. Ancak aynı küme içinde iki sunucu birden devre dışı kalırsa bu fiziksel sunucular üzerinde çalışan sanal sunuculara erişim kesilmiş olur. Ta ki bu fiziksel sunucular tekrar çalışır duruma getirilene kadar bu sanal sunuculara erişim sağlanamaz.

North America		Europe		Asia Pacific						
		⏪	Apr 25	Apr 24	Apr 23	Apr 22	Apr 21	Apr 20	Apr 19	⏩
Amazon CloudFront			✔	✔	✔	✔	✔	✔	✔	
Amazon CloudWatch (N. California)			✔	✔	✔	✔	✔	✔	✔	
Amazon CloudWatch (N. Virginia)			✔	✔ ²	✔ ²	✔ ²	✔ ²	✔	✔	
Amazon EC2 (N. California)			✔	✔	✔	✔	✔	✔	✔	
Amazon EC2 (N. Virginia)			✔	⚠	✖	✖	✖	✔	✔ ²	

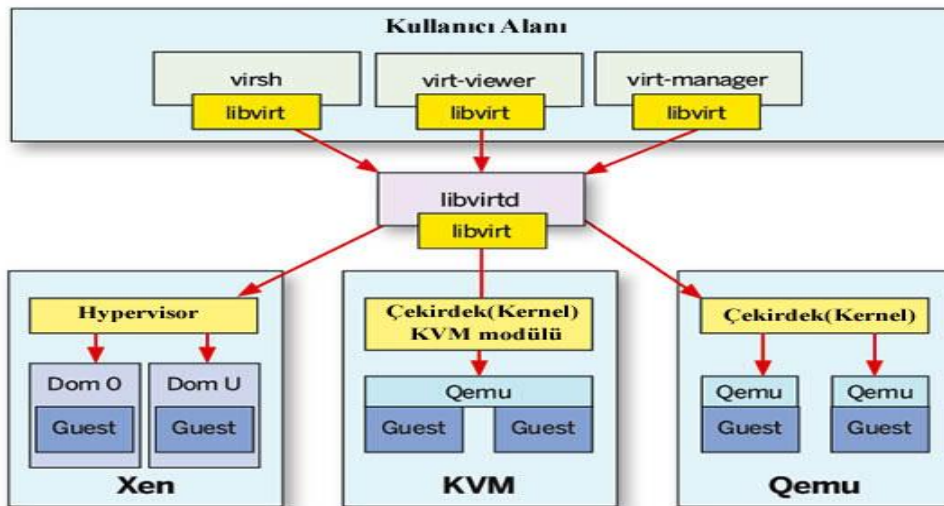
Şekil.5. Amazon EC2 servis kesintisi.

3.3. Sunucu Sanallaştırma

Sanallaştırma genel olarak bilgisayar kaynaklarının kullanıcılardan soyutlanması, fiziksel yapının mantıksal yapıya dönüştürülmesi anlamına gelir. Fiziksel yapı, ihtiyacımız doğrultusunda istediğimiz kadar mantıksal parçalara bölerek, toplam sunucu verimliliğini optimize etmeye, mevcut iş yüklerini sanal bir katmana taşıyarak bir sunucuda birden fazla işletim sistemi üzerinde yine birden fazla iş yükü çalıştırmaya yarar. Bu yapı üzerinde oluşturulan sanal sunucuda işletim sistemi tamamen fiziksel bir makinedeymiş gibi çalışır. Kendilerine ait işlemci çekirdekleri, bellekleri, depolama birimleri, ağ erişimlerine sahiptirler.

Sanallaştırmanın avantajları çok uzun süredir bilinmesine rağmen sunucu teknolojileri son birkaç yılda ilerlediği için artık çok verimli bir seçenek haline gelmiştir. Güçlü sunucular ucuza mal edilebilmekte, böylece tek bir sunucuda birçok işletim sistemi birbirinden bağımsız güvenli bir biçimde çalışarak ciddi tasarruflar sağlanabilmektedir.

Sanallaştırma, enerji verimliliğinin yanı sıra, daha az alanda daha fazla sunucu çalıştırılabilme imkanı sağlamakta ve sunucuların yönetiminin daha kolay yapılmasına olanak sağlayarak toplam sahip olma maliyetlerini aşağı çekmektedir.



Şekil.6. Sanallaştırma görünümü.

Bulut sağlayıcılarının sektörde rekabet içinde olabilmeleri için avantajlı fiyatlar sunmaları önemlidir. Bunu sağlamak açısından klasik bulut sistemleri yerine yazılımın daha çok yer aldığı, yaygın donanımların kullanıldığı (SATA, SSD maliyet farkı gibi) yapıları oluşturmaları gerekmektedir. Özel depolama cihazlarını (SAN) kullanmayı tercih etmeyip yazılımsal olarak bu sistemlerin yapacaklarını da kapsayacak yapılar oluşturmak veya kullanmayı tercih etmek maliyet açısından avantaj sağlayıp son kullanıcıya bu fiyat avantajı yansıtılması gerekir. Diğer yandan oluşturulan bulut sisteminde alt yapıya hakimiyet önem taşımaktadır, çünkü bulut yapısında kaybedilen her bir dakika önemlidir. Muhtemel karşılaşılabilecek sorunların otomatize edilip bir otokontrol yapısı oluşturulması, son kullanıcıya herhangi bir sorunun yansıtılmadan çözülmesi, yedeklilik yapılarının oluşturulması kritik öneme sahiptir.

İnternet alt yapısının yedekli olması, birbirinden farklı internet sağlayıcılarıyla çalışmak, elektrik sisteminin alt yapısı, güçlü jeneratör desteği ve yedeği, ısıtma ve soğutma özellikleri ve diğer gerekli fiziksel özelliklerde göz önünde bulundurulmalıdır. Bu yapı üzerinde bir çok küçük, orta ve büyük firma hizmet almaktadır.



Şekil.7. Veri merkezinde barındırılan sunucular.

3.4. Kümelenmiş (Clustered) Yapı

Cluster birden fazla sunucunun kümelenerek tek bir sunucu gibi davranmasıdır. Bu şekilde yüksek erişilebilirlik, yedeklilik sağlanmış olmaktadır. Bu ortamda sunucular tarafından kullanılan veriyi ortak bir disk alanında saklamak ve bu ortak diske aynı anda cluster sisteminin yalnızca bir üyesinin erişimini sağlanması şeklinde çalışmaktadır. Kümelenmiş sunucular fiziksel kablolarla ve yazılımlarla bağlanır.

Sunucu üzerindeki lokal bir diski depolama olarak tanımlayabiliriz. Kümelenmiş yapılarında bahsettiğimiz depolama ise sunucuların ortak olarak erişebildiği depolama alanları sağlayan cihazlar/yazılımlardır.

Okuma/yazma performansını etkileyen noktaların başında depolama üzerindeki disklerdir. Genelde Sata, SAS, SCSI diskler kullanılır. Bu fiziksel disklerin RPM değerleri ve bağlı bulundukları kontroller kartların donanımsal özellikleri de okuma/yazma performansına direkt etki etmektedir.

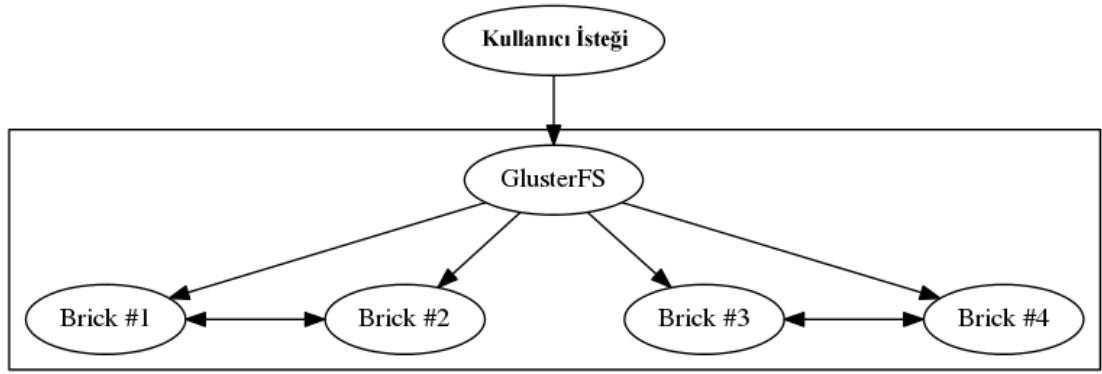
Failover(Yük Devretme) Cluster yapısında organizasyonumuz içerisindeki sunucuların ve bu sunucular üzerinde çalışan çeşitli uygulamaların/servislerin hata ile karşılaşması durumunda durumunda (genelde donanımsal başarısızlıklar yüzünden), hizmet veremez duruma geçen ilgili servis ve uygulamaların cluster üyesi diğer sunucu/sunucular üzerinde hizmet vermeye devam etmesini sağlayan bir teknolojidir.

Failover Cluster yapısı için genelde en az iki sunucu olmak zorundadır. Bununla birlikte çoğu zaman node'ların erişebildiği ortak bir depolamada gereklidir.

Sanallaştırmanın temel amaçlarından olan çok sayıdaki fiziksel sunucuyu daha az sayıdaki fiziksel sunucu üzerinde sanal olarak konumlandırma işlemi sonucunda, aynı fiziksel makine üzerinde birden fazla sanal iş yükü çalışmaya başlar ve bu makinenin bir şekilde donanımsal veya yazılımsal sorunla karşılaşp devre dışı olması durumunda

üzerindeki tüm iş yükleri bu durumdan olumsuz etkilenir. Yani cluster ile güvenceye alınmamış ortamlarda çalışan sanal sistemlerde risk çok daha büyüktür.

Cluster ile amaçlanan, beklenmedik bir anda meydana gelen arızalara ve genelde donanımsal sorunlar yüzünden yaşanan servis kesintilerine karşı önlem almak, bu doğrultuda fiziki makine üzerindeki yükü otomatik olarak diğer cluster üyesi sunucular üzerinden hizmete sunabilmektir. Günümüzde yaşanan acımasız rekabet ortamında en büyük hedef kesintisiz hizmetler sunabilmektir. Kesintisiz servis sunabilmek şirketin rakiplerine karşı avantajlı duruma geçmesini sağlayacaktır.



Şekil.8. Genel depolama yapısı.

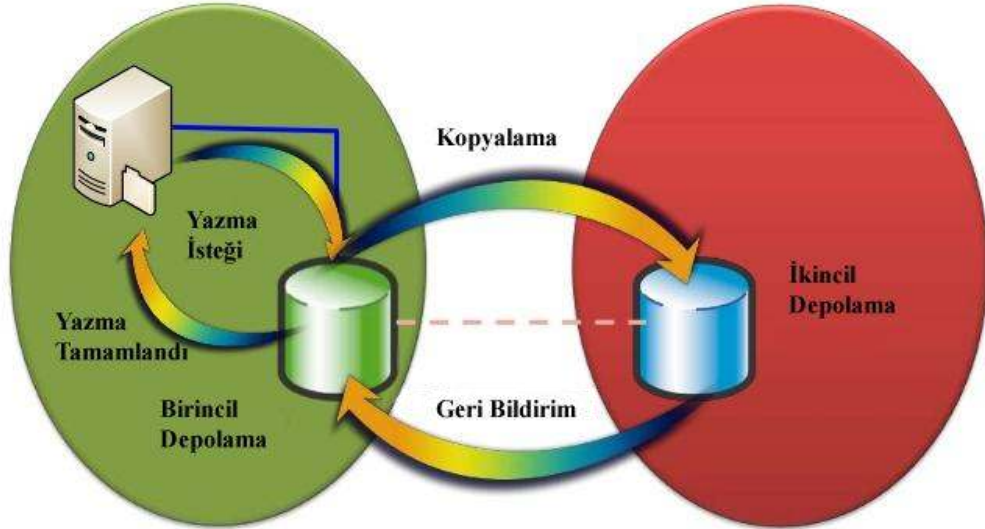
3.5. Veri Replikasyonu

Genel anlamıyla replikasyon aynı kaynağın bir veya birden fazla kopyasının farklı farklı sunucularda saklanmasıdır. Ana sunucunun devre dışı kalması durumunda replike edilmiş sağlıklı bir kopyanın bulundurulması sayesinde veri kaybının önlenmesi ve sistemin çalışır halde tutulması amaçlanır. Kullanıcı sanal sunucusunu bu fiziki sunuculardan birinde çalıştırır. Herhangi bir fiziksel sunucudaki datalar diğer fiziksel sunucularla senkronizedir. Senkronize replikasyon "sıfır veri kaybı" garanti eder. Bunu atomik yazma işlemi vasıtasıyla gerçekleştirir. Yazma işlemini ya iki tarafta tamamlar veya bu işlemi hiç gerçekleştirme mantığını kullanır. Bu işlemi gerçekleştirirken işlem tamamlandı bilgisini onaylama gelmediği müddetçe kabul etmez. Senkronize

replikasyonda ilk olarak sunucudan gelen veri kaynak depolama birimine yazılır ancak onaylama paketi sunucuya geri iletilmez. İkinci adımda hedef veri depolama birimine yazılır ve hedef veri depolama birimi onaylama paketini kaynağa gönderir. Son adımda kaynak depolama birimi sunucuya onaylama paketini gönderir ve veri yazılmış olur. Burada RPO süresi 0 olur. RPO mevcut yapının hedefteki yapıyla eşlemesine, yani olası bir felaket anında riske edilen data miktarını yansıtan süreye denir. RTO kesintiye uğrayan iş sürecinin ne kadar süre sonra çalışılır hale getirileceğine dair hedef süredir. [7] RTO süresi kullanılan teknolojiye göre değişiklik gösterebilir. Burada ki ara teknoloji çok önemlidir. Çünkü diğer lokasyona erişim sırasında bir sorun olursa diğer tarafta verinin bozulmasına sebep olabilir.

Veriler kaynak birimden hedef birime anlık olarak ve blok bazında gönderilir. Kaynak birimden hedef birime seçtiğiniz bütün dosyalar gönderildikten sonra değişen dosyaların sadece değişen blokları gönderilir.

Host: Yazma işlemini ya iki tarafta tamamla veya bu işlemi hiç gerçekleştirme

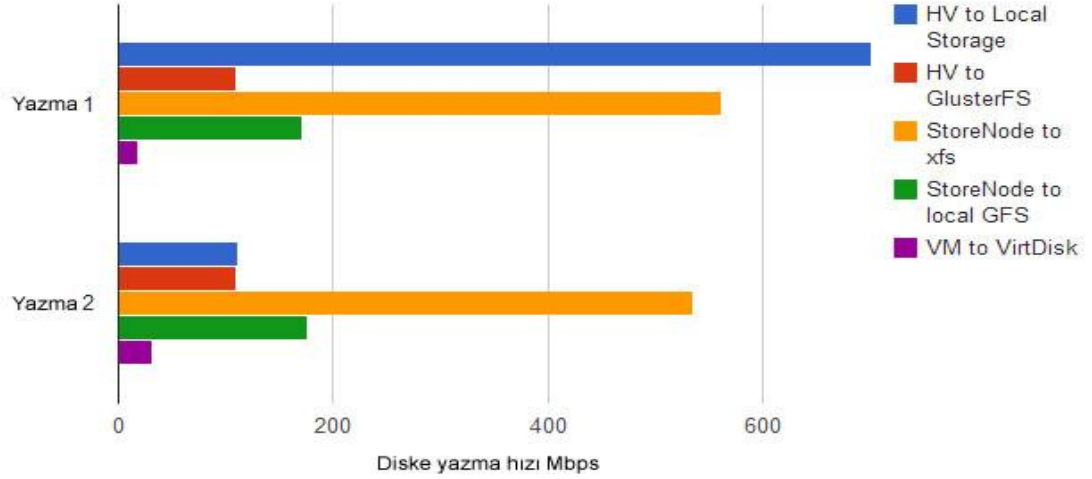


Şekil.9. Veri replikasyonu senkronize çalışma mantığı.

4. PERFORMANS DEĞERLENDİRMESİ

4.1. Etkin ve Verimlilik

Lokal depolama ile network üzerinden depolama ile ilgili karşılaştırmak gerekirse;



Şekil.10. Depolama karşılaştırması.

GlusterFS network üzerinde kendinize ait NAS cihazı gibi kullanabilir.[8] NAS bir depolama ürünüdür. Günümüzde sık kullanılan bir yöntemdir. Ağa Bağlı Depolama (Network Attached Storage - genel olarak NAS olarak kısaltılır) sabit disk depolama cihazlarıdır. Bir ağdaki birden fazla bilgisayarın, aynı anda aynı depolama alanını paylaşımlarına olanak tanırırlar. GlusterFS'in NAS'dan fazlalığı kümelenendirilmiş bir yapıda olmasıdır. GlusterFS, hataya dayanıklı ve ölçeklenebilir ağ depolama sağlamak için tasarlanmış bir dosya sistemidir. En önemli yanı tek bir hata noktası (single point of failure) olmamasıdır, yani sistemin bir kısmında bir arıza meydana gelirse ve sistemden çıkarılırsa tüm sistem çalışmaya devam eder.

Bu sistem;

- Yüksek erişilebilir bir depolama oluşturulmasında,
- CDN olarak kullanımında,
- Felaket kurtarma durumlarında kullanılabilir.
- Fault Tolerant yani sistem üzerinde oluşan hataların sistemi çökertmemesi sağlanır.

Yüksek Erişilebilirlik

Bu demek oluyor ki sistem her durumda çalışmak üzere dizayn edilmiştir. Çalıştırılan sanal sunucuların birden fazla sunucu üzerine yayarak, herhangi bir sunucu üzerinde oluşabilecek hatadan dolayı, çalışan sistemin etkilenmemesi için kullanılmaktadır.

Ölçeklendirilebilirlik

Mevcut sistem üzerinde herhangi bir değişiklik yapılmadan ve çalışan sistemin devamlılığını bozmadan büyütmemize veya küçültmemize olanak sağlayan özelliktir. Sanal sunucu çalışır durumdayken bellek, işlemci ve disk gibi kaynaklar üzerinde değişiklik yapmamıza imkan tanımaktadır.

4.1.1. Bulut Kurulumu

4.1.1.1. Mimari Yapı Hakkında

OpenStack'i fiziksel sunucu olarak varsayacağımız iki adet sanal sunucu üzerine kuracağız.

1. sunucu(Node 1):

- Bulut yönetim kontrolünü sağlayacak. Üzerinde nova-api, nova-scheduler, nova-network, glance, horizon, MySQL, ve RabbitMQ çalışıyor olacak.
- Ölçeklenebilirliği ve depolamayı sağlayacak. (nova-compute, qemu, nova-volume)

2. sunucu(Node 2):

- Ölçeklenebilirliği ve depolamayı sağlayacak. (nova-compute, qemu, nova-volume)

Buradan anlaşılacağı gibi makineler ya node1 ya da node2 üzerinde oluşacaktır(nova-scheduler makinenin nerede oluşturulacağına karar verir). Node2 aktif olmasa bile yeni makine oluşturulabilir. Node1 tek başına çalışacaktır.

Nova-volume'u node2 üzerine de yüklersek, nova-scheduler oluşturulan makinenin hangi sunucu diskinde olacağına karar verir(node1 veya node2). İlk etapta nova-scheduler rastgele karar verir, ancak sonrasında sunucuların diskleri üzerinde kullanılabilir alana göre karar verir.

Genel kurulum akabinde Horizon (Gösterge paneli web arayüzü) projemize dahil edeceğim, Horizon Django kütüphanesi kullanılarak yazılmıştır. Horizon hemen hemen herşeyi yönetmemizi sağlayacaktır, arka planda programlama arayüzü(API) kullanarak OpenStack öğeleriyle iletişimi sağlar.

Gösterge Paneli;

- Proje Paneli: sunucular, depolar, genel ip, güvenlik grupları, yedekleme
- Yönetici Paneli: sunucular, servisler, paketler, imajlar, projeler, kullanıcılar ve kota bölümünden oluşur.

4.1.1.2. Kurulum için Seçilen Yazılım Ve Versiyonları

- İşletim Sistemi: Linux Ubuntu Server 12.04, 64 bits.
- Platform: OpenStack 2012.2 Folsom, Nova, Glance, Keystone ve Horizon.

4.1.1.3. Donanım Gereksinimleri

- Node1: 1 Harddisk, 1 Ağ Kartı
- Node2: 1 Harddisk, 1 Ağ Kartı

4.1.1.4. Ağ Yapılandırması

- Ağ Türü = VLAN(VlanNetworkManager)

Eth0(public/external)

Network 1: 192.168.1.0/24

Yönetici ağı bulutta kullanılan öğeler ile MySQL veritabanı ve RabbitMQ mesaj servisi arasında iletişimi sağlar.

Depolama ağı iSCSI depolama trafiği için kullanılıp, depolama birimleri ile kullanılan fiziki sunucuların arasında kullanılır.

4.1.1.5. Bulut Kurulum Adımları

Sunucu Yüklemeleri (Yönetici Sunucu)

- RabbitMQ (mesaj/sıraya alma)

```
root@ubuntu:~# apt-get install rabbitmq-server
```

Şekil.11. RabbitMQ yükleme işlemi.

MySQL Veri Tabanı (Veri Tabanı Sunucu)

MySQL veritabanı şifresi olarak 123456 seçiyorum.

```
root@ubuntu:~# apt-get install mysql-server
```

Şekil.12. MySQL yükleme işlemi.

MySQL'e diğer sunuculardan erişilebilmesini sağlayalım.

```
root@ubuntu:~# sed -i 's/127.0.0.1/0.0.0.0/g' /etc/mysql/my.cnf
```

Şekil.13. MySQL dışarıdan erişim sağlanması.

MySQL servisini yeniden başlatalım.

```
root@ubuntu:~# service mysql restart
```

Şekil.14. MySQL servisinin yeniden başlatılması.

Nova, Glance ve Keystone aynı sunucudaki MySQL veritabanını kullanırlar fakat veritabanı isimleri ayrı olacak.

Nova Yüklemesi (Bulut Servisi)

Nova veri tabanı ve nova kullanıcı adı oluşturalım.

```
root@ubuntu:~# mysql -uroot -p123456 -e 'CREATE DATABASE nova;';mysql -uroot -p123456 -e "GRANT ALL PRIVILEGES ON . TO 'nova'@'%' WITH GRANT OPTION;";mysql -uroot -p123456 -e "SET PASSWORD FOR 'nova'@'%' = PASSWORD('123456');";
```

Şekil.15. Nova servisinin kurulumu.

Nova Bileşenlerinin Yüklemesi

Yönetici birimi:

```
root@ubuntu:~# apt-get install nova-api nova-scheduler nova-network
```

Şekil.16. Nova bileşenlerinin kurulumu.

Depolama birimi:

```
root@ubuntu:~# apt-get install nova-volume
```

Şekil.17. Nova depolama servisi kurulumu.

Bulut birimi(ölçeklenebilirlik, sanallaştırma işlemleri, çeşitli sürücüler):

```
root@ubuntu:~# apt-get install nova-compute
```

Şekil.18. Nova sanallaştırma servisi kurulumu.

Nova-compute yüklemesi yapıldığında KVM/Qemu ve libvirt otomatik yüklenir.

Tüm nova bileşenleri listesi aşağıdaki gibidir. Bu paketler otomatik olarak yüklenmiştir.

nova-api: nova-common, python, upstart-job.

nova-scheduler: nova-common, python, upstart-job.

nova-network: netcat, vlan, bridge-utils, dnsmasq-base, iputils-arping, dnsmasq-utils, nova-common, python2.7, upstart-job.

nova-compute: lsb-base, nova-common, qemu-utils, kpartx, curl, parted, vlan, ebttables, gawk, iptables, open-iscsi, nova-compute-kvm, python2.7, upstart-job.

nova-compute-kvm: nova-compute, python-libvirt, libvirt-bin, kvm.

nova-compute-lxc: nova-compute, python-libvirt, libvirt-bin.

nova-volume: nova-common, lvm2, tgt, python2.7, upstart-job.

python-nova:python2.7, openssh-client, openssl, python-boto, python-m2crypto, python-pycurl,python-daemon, python-carrot, python-kombu, python-lockfile, python-gflags, python-libxml2, python-ldap, python-sqlalchemy, python-eventlet, python-routes, python-webob, python-cheetah, python-netaddr, python-paste,python-pastedeploy, python-tempita, python-migrate, python-glance, python-novaclient, python-simplejson, python-lxml, python-feedparser, python-xattr, python-suds, python-iso8601, sudo.

nova-common: python-amqpplib, python-nova, python, adduser.

Eğer KVM yerine LXC kullanılabilir:

```
root@ubuntu:~# apt-get remove nova-compute-kvm;apt-get install nova-compute-lxc
```

Şekil.19. Nova lxc sanallaştırma servisi kurulumu.

Konfigürasyon dosyasında seçili libvirt türünü değiştirmek gerekli.

```
root@ubuntu:~# nano /etc/nova/nova.conf
```

Şekil.20. Nova konfigürasyon dosyası düzenlemesi.

```
# COMPUTE
connection_type=libvirt
compute_driver=libvirt.LibvirtDriver
libvirt_type=lxc
```

Şekil.21. Nova sanallaştırma türü değiştirilmesi.

Ardından nova-compute servisini yeniden başlatalım.

```
root@ubuntu:~# service nova-compute restart
```

Şekil.22. Nova sanallaştırma servisi yeniden başlatılması.

Eğer herhangi bir nova servisini devre dışı bırakmak istersek;

```
root@ubuntu:~# nova-manage service disable -host=node1 -service=nova-compute
```

Şekil.23. Nova servisinin devre dışı bırakılması.

Nova-compute servisi artık node1 birimde çalışmayacaktır.

Glance Yüklemesi (İmaj Modülü)

Glance veri tabanı ve glance kullanıcı adı oluşturalım.

```
root@ubuntu:~# mysql -uroot -p123456 -e 'CREATE DATABASE glance;';mysql -uroot -p123456 -e "GRANT ALL PRIVILEGES ON . TO ' glance'@'%' WITH GRANT OPTION;";mysql -uroot -p123456 -e "SET PASSWORD FOR ' glance'@'%' = PASSWORD('123456');";
```

Şekil.24. Glance veri tabanı işlemleri.

```
root@ubuntu:~# apt-get install glance
```

Şekil.25. Glance modülü kurulumu.

Keystone Yüklemesi (Kimlik Modülü)

Keystone veri tabanı ve keystone kullanıcı adı oluşturalım.

```
root@ubuntu:~# mysql -uroot -p123456 -e 'CREATE DATABASE keystone;';mysql -uroot -p123456 -e "GRANT ALL PRIVILEGES ON . TO ' keystone'@'%' WITH GRANT OPTION;";mysql -uroot -p123456 -e "SET PASSWORD FOR ' keystone'@'%' = PASSWORD('123456');";
```

Şekil.26. Keystone veri tabanı işlemleri.

```
root@ubuntu:~# apt-get install keystone
```

Şekil.27. Keystone modülü kurulumu.

Sunucu Yüklemeleri (Basit Sunucu)

```
root@node2:~# apt-get install nova-compute
```

Şekil.28. Node2 sanallaştırma servisi kurulumu.

Eğer birden fazla basit sunucu var ise diğer sunucu yüklemeleri aynı şekilde yapılır:

```
root@node3:~# apt-get install nova-compute
```

Şekil.29. Node3 sanallaştırma servisi kurulumu.

Sunucu Yapılandırmaları

Her iki sunucunun hosts dosyalarını:

```
root@ubuntu:~# nano /etc/hosts
```

Şekil.30. Host dosyası düzenleme komutu.

Aşağıdaki şekilde düzenleyelim.

```
127.0.0.1      localhost.localdomain  localhost
192.168.10.156 server1.example.com      ubuntu
192.168.10.195 server2.example.com      ubuntu2
```

Şekil.31. Host dosyası konfigürasyonu.

Ağ arayüzünü aşağıdaki şekilde düzenleyelim.

1.sunucu için:

```
root@node1:~# nano /etc/network/interfaces
```

Şekil.32. Ağ arayüzü düzenleme işlemi.

Aşağıdaki şekilde düzenleyelim.

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
#NetworkManager
iface eth0 inet static
    address 192.168.10.156
    netmask 255.255.255.0
    broadcast 192.168.10.255
    gateway 192.168.10.2
```

Şekil.33. Node1 ağ arayüzü ayarları.

Statik ip kullanılmasının sebebi bu makineye tüm diğer fiziki sunuculardan erişim sağlanacaktır.

2.sunucuda:

```
root@node2:~# nano /etc/network/interfaces
```

Şekil.34. Node2 ağ arayüzü düzenleme işlemi.

Aşağıdaki şekilde düzenleyelim.

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
    address 192.168.10.195
    netmask 255.255.255.0
    broadcast 192.168.10.255
    gateway 192.168.10.2
```

Şekil.35. Node2 ağ arayüzü ayarları.

Nova Konfigürasyon Dosyası Yapılandırması

```
root@ubuntu:~# nano /etc/nova/nova.conf
```

Şekil.36. Nova konfigürasyon dosyası komutu.

```

GNU nano 2.2.6                                File: /etc/nova/nova.conf

[DEFAULT]

# LOG/STATE
verbose=True
logdir=/var/log/nova
state_path=/var/lib/nova
lock_path=/var/lock/nova
logging_context_format_string=%(asctime)s %(color)s%(levelname)s %(name)s [%(request_id)s %(user_name)s %(process_id)s]
logging_default_format_string=%(asctime)s %(color)s%(levelname)s %(name)s [-%(color)s] %(instance)s%(color)s%%
logging_debug_format_suffix=from (pid=%(process)d) %(funcName)s %(pathname)s:%(lineno)d
logging_exception_prefix=%(color)s%(asctime)s TRACE %(name)s %(instance)s

# AUTHENTICATION
use_deprecated_auth=false
auth_strategy=keystone
keystone_ec2_url=http://192.168.10.156:5000/v2.0/ec2tokens

# SCHEDULER
scheduler_driver=nova.scheduler.simple.SimpleScheduler
#compute_scheduler_driver=nova.scheduler.filter_scheduler.FilterScheduler

# DATABASE
sql_connection=mysql://nova:nova@192.168.10.156/nova_db

# NETWORK
dhcpbridge_flagfile=/etc/nova/nova.conf
dhcpbridge=/usr/bin/nova-dhcpbridge
network_manager=nova.network.manager.VlanManager

# NETWORK
dhcpbridge_flagfile=/etc/nova/nova.conf
dhcpbridge=/usr/bin/nova-dhcpbridge
network_manager=nova.network.manager.VlanManager
public_interface=eth0
vlan_interface=eth0
fixed_range=10.10.10.0/24
routing_source_ip=192.168.10.156
force_dhcp_release=True
iscsi_helper=ietadm
iscsi_ip_address=192.168.10.156
my_ip=192.168.10.156
#multi_host=True
#firewall_driver=nova.virt.libvirt.firewall.IptablesFirewallDriver

# NOVNC
novnc_enabled=true
novncproxy_base_url=http://192.168.10.156:6080/vnc_auto.html
novncproxy_port=6080
xvpvncproxy_base_url=http://192.168.10.156:6081/console
vncserver_listen=0.0.0.0
vncserver_proxyclient_address=127.0.0.1

# APIs
osapi_compute_extension=nova.api.openstack.compute.contrib.standard_extensions
s3_host=192.168.10.156
ec2_host=192.168.10.156
ec2_dmz_host=192.168.10.156
#cc_host=192.168.10.156
enabled_apis=ec2,osapi_compute,metadata
nova_url=http://192.168.10.156:8774/v1.1/
ec2_url=http://192.168.10.156:8773/services/Cloud
#volume_api_class=nova.volume.cinder.API

# RABBIT
rabbit_host=192.168.10.156

# GLANCE
glance_api_servers=192.168.10.156:9292
image_service=nova.image.glance.GlanceImageService

# COMPUTE
connection_type=libvirt
compute_driver=libvirt.LibvirtDriver
libvirt_type=qemu
libvirt_cpu_mode=host-model
allow_resize_to_same_host=True
libvirt_use_virtio_for_bridges=true
start_guests_on_host_boot=false

# CINDER
volume_api_class=nova.volume.cinder.API
osapi_volume_listen_port=5900

```

Şekil.37. Nova konfigürasyon dosyası.

-my_ipparametresi (makinenin IP adresi) belirtilmek zorundadır. Burada 192.168.10.156 olarak belirtilmiştir. İkinci makinemiz birinci makine ile bu ip üzerinden iletişim kuracaktır.

- libvirt_type parametresi kullanılan sanallaştırma teknolojisi türünü belirtir, projemizde qemu kullanıyoruz.

- Eğer Qemu OpenStack sanal makineye veya bir fiziki makineye yüklendiğinde CPU donanımı sanallaştırma teknolojisini desteklememektedir. Örn: Intel VT-x donanımında kullanılabilir. Oldukça yavaş çalışır. X86 mimarisi hariç iletişim kurabilir. Buna örnek ARM mimarisi gösterilebilir.

OpenStack ile desteklenen sanallaştırma teknolojileri:

- KVM (via libvirt)
- Qemu (via libvirt)
- LXC (via libvirt)
- UML (via libvirt)
- XCP (Xen)
- XenServer (from Citrix)
- ESX (from VMware)
- Hyper-V (from Microsoft)

Glance Dosyası Yapılandırması

```
root@ubuntu:~# nano /etc/glance/glance-registry.conf
```

Şekil.38. Glance kayıt dosyası düzenleme.

Parametresini aşağıdaki gibi değiştirelim.

```
# SQLAlchemy connection string for the reference implementation
# registry server. Any valid SQLAlchemy connection string is fine.
# See: http://www.sqlalchemy.org/docs/05/reference/sqlalchemy/connections.
sql_connection = mysql://glance:glance@192.168.10.156/glance_db
```

Şekil.39. Glance kayıt dosyası konfigürasyonu.

Keystone Dosyası Yapılandırması

```
root@ubuntu:~# nano /etc/keystone/keystone.conf
```

Şekil.40. Keystone konfigürasyon dosyası.

Parametreleri aşağıdaki gibi değiştirelim.

```
[DEFAULT]
# A "shared secret" between keystone and other openstack services
admin_token = 012345SECRET99TOKEN012345

[sql]
# The SQLAlchemy connection string used to connect to the database
connection = mysql://keystone:keystone@192.168.10.156/keystone_db

[identity]
driver = keystone.identity.backends.sql.Identity
```

Şekil.41. Keystone konfigürasyon dosyası yapılandırması.

API uç noktaları (endpoints) varsayılan şekliyle kullanıyoruz.

Glance ve Nova Yapılandırması

```
root@node1:~# nano /etc/nova/api-paste.ini
root@node1:~# nano /etc/glance/glance-api-paste.ini
root@node1:~# nano /etc/glance/glance-registry-paste.ini
```

Şekil.42. Glance ve Nova konfigürasyon dosyaları.

Parametrelerini aşağıdaki gibi değiştirelim.

```
admin_tenant_name = admin
admin_user = admin
admin_password = password
admin_token = 012345SECRET99TOKEN012345
```

1. sunucuyu yeniden başlatalım. (Node1)

```
root@ubuntu:~# reboot
```

Şekil.43. Sunucuyu yeniden başlatma işlemi.

Veri tabanında tabloları oluşturalım:

```
root@node1:~# nova-manage db sync
```

Şekil.44. Nova veri tabanı tabloları oluşturma işlemi.

```
root@node1:~# glance-manage version_control 0; glance-manage db_sync;keystone-manage db_sync;
```

Şekil.45. Glance ve keystone veri tabanı tabloları oluşturma işlemi.

Sadece 2. sunucuda 1. sunucudaki nova.conf dosyasını 2.sunucuya kopyalayalım.

```
root@node1:~# scp root@node1:/etc/nova/nova.conf /etc/nova
```

Şekil.46. Dosya kopyalama işlemi.

my_ip parametresini 2.sunucunun ip adresi olacak şekilde değiştirelim.

```
root@ubuntu:~# nano /etc/nova/nova.conf
```

Şekil.47. Nova konfigürasyon dosyası komutu.

```
my_ip=192.168.10.195  
#multi_host=True  
#firewall_driver=nova.virt.libvirt.firewall.IptablesFirewallDriver
```

Şekil.48. Nova konfigürasyon dosyası düzenleme işlemi.

Her iki sunucuyu yeniden başlatalım.

```
root@node1:~# reboot  
root@node2:~# reboot
```

Tüm servislerin çalışıyor olduğundan emin olalım:

```
root@node1:~# nova-manage service list
```

Aşağıdaki gibi bir çıktı gözükecektir.

```

root@ubuntu:~# nova-manage service list
Binary      Host      Zone      Status
State Updated_At
nova-consoleauth ubuntu    nova      enabled
:-) 2014-01-12 19:56:00
nova-network ubuntu    nova      enabled
XXX 2014-01-12 20:04:45
nova-compute ubuntu    nova      enabled
:-) 2014-01-12 19:56:00
nova-scheduler ubuntu    nova      enabled
:-) 2014-01-12 19:56:02
nova-cert ubuntu    nova      enabled
:-) 2014-01-12 19:55:59
nova-consoleauth ubuntu2    nova      enabled
:-) 2014-01-12 19:56:17
nova-compute ubuntu2    nova      enabled
:-) 2014-01-12 19:56:19
nova-console ubuntu2    nova      enabled
:-) 2014-01-12 19:56:20
root@ubuntu:~# █

```

Şekil.49. Servislerin kontrol edilmesi.

Tüm yapılandırma parametrelerini kontrol etmek için:

```

root@ubuntu:~# nova-manage config list █

```

Şekil.50. Konfigürasyon ayarlarının listelenmesi.

Eğer herhangi bir hata oluşursa, /var/log/nova yolundaki günlük dosyaları kontrol edilebilir.

Ağ Yapılandırması (nova-network)

VLAN ağ modelinde her bir oluşturulan projenin kendine ait özel bir VLAN/alt ağı vardır. Projeler için VLAN/ alt ağı:

```

root@node1:~# nova-manage network create --label vlan1 --fixed_range_v4 10.0.1.0/24 --num_networks 1 --network_size 256 --vlan 1 █

```

Şekil.51. Sanal ağ oluşturulması.

Örneğin vlan1 her zaman birinci projenin oluşturacağı sanal makinelerde kullanılır. Eğer farklı bir proje kullanıcısından sanal makine oluşturulursa vlan1 bu makine için kullanılamaz. Ayrıca bu projeye ait bir alt vlan yok ise sanal sunucu oluşmayıp hataya düşecektir.

Dışarıya açık ip (public ip) oluşturmak için:

```
root@node1:~# nova-manage floating create --ip_range 192.168.10.164/27
```

Şekil.52. Dışa açık ip oluşturulması.

Ip tablosu (iptables) esnek ipleri yapılandırmak için ağ birimde bulunur (1. sunucu burada). Ip tablosu ayrıca güvenlik duvarı kurallarını da yapılandırmak için kullanılmıştır. (1. ve 2. sunucu)

Depolama Yapılandırması (nova-volume veya cinder)

```
root@node1:~# nano /etc/cinder/api-paste.ini
```

Parametrelerini aşağıdaki gibi değiştiririm.

```
sed -i "s/127.0.0.1/$IP/g" /etc/cinder/api-paste.ini
sed -i "s/%SERVICE_TENANT_NAME%/service/g" /etc/cinder/api-paste.ini
sed -i "s/%SERVICE_USER%/cinder/g" /etc/cinder/api-paste.ini
sed -i "s/%SERVICE_PASSWORD%/cinder/g" /etc/cinder/api-paste.ini
```

/etc/cinder/cinder.conf dosyasını güncelleştirelim.

```
root@node1:~# nano /etc/cinder/cinder.conf
sql_connection = mysql://cinder:cinder@ 192.168.10.156/cinder_db
```

Depolama servislerini yeniden başlatalım.

```
root@ubuntu:~# service cinder-volume restart; service cinder-api restart;
```

Şekil.53. Depolama servisi yeniden başlatma işlemi.

Kontrol Paneli Yapılandırması (Horizon)

```
root@node1:~# apt-get install openstack-dashboard
```

Şekil.54. Kontrol paneli yükleme işlemi.

Gerekli paketleri de yükleyelim.

```
root@node1:~# apt-get install memcached
```

Şekil.55. Önbellekleme yükleme işlemi.

Ubuntu temasını yükleme:

```
root@node1:~# apt-get install openstack-dashboard-ubuntu-theme
```

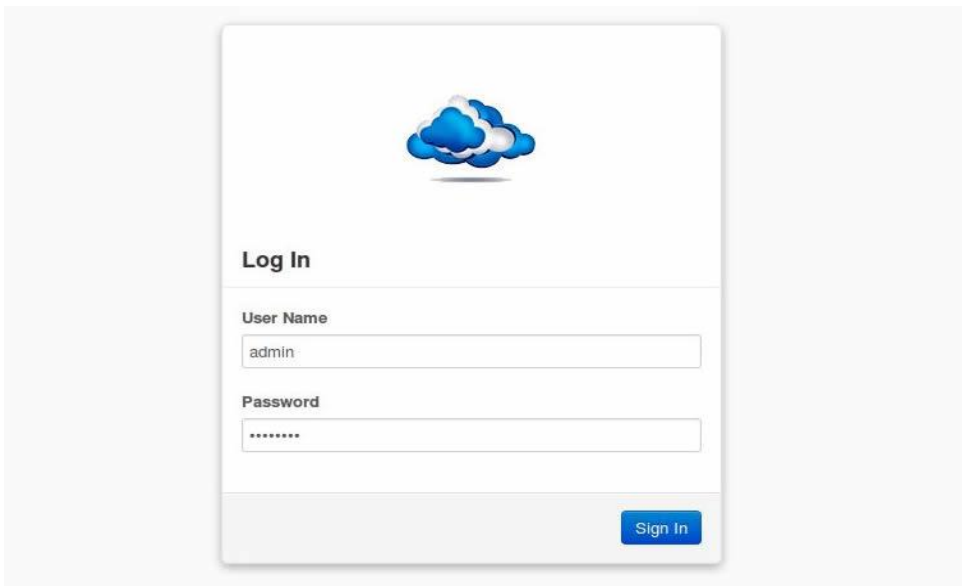
Şekil.56. Ubuntu tema yükleme.

```
root@node1:~# service apache2 restart
```

Şekil.57. Apache servisi yeniden başlatma işlemi.

Şimdi dışa açık ip adresimizi girerek kontrol paneline erişebiliriz.

<http://192.168.10.156>



Şekil.58. Kontrol paneli giriş ekranı.

Kullanıcı adı: admin

Şifre: password

4.1.2. Yüksek Erişilebilir Depolama Kurulumu

İki fiziksel sunucuya yüksek erişilebilirlik sağlayan bir depolama sistemi kurmak, sistemimizin üzerinde çalışan sanal sunucuların devamlılığı açısından kritik bir öneme sahiptir. Her depolama sunucusu, diğer depolama sunucusunun bir aynası olacak, ayrıca içerdiği veriler birbirleriyle otomatik olarak kopyalanacaktır.

Her iki sisteminde birbirlerinin host adlarını çözümlemesi gerekmektedir. Bunun için host dosyasını aşağıdaki gibi değiştirelim;

```
127.0.0.1      localhost
192.168.10.156 server1.example.com  node1
192.168.10.195 server2.example.com  node2
```

Her iki sistemde glusterfs-server yüklemelerini gerçekleştirelim.

```
apt-get update
apt-get install python-software-properties
add-apt-repository ppa:semiosis/ubuntu-glusterfs-3.4
apt-get update
apt-get install glusterfs glusterfs-fuse glusterfs-server
```

Sunucuları güvenli depolama havuzuna ekleyelim(Trusted Storage Pool).

```
root@node1:~# gluster peer probe server2.example.com veya gluster peer
probe 192.168.10.195
```

```
root@node2:~# gluster peer probe server1.example.com veya gluster peer
probe 192.168.10.156
```

```
root@ubuntu:~# gluster peer probe server2.example.com
peer probe: success: host server2.example.com port 24007
root@ubuntu:~#
root@ubuntu:~#
```

Şekil.59. Sunucuların depolamaya eklenmesi.

```
root@ubuntu:~# gluster peer status
Number of Peers: 1

Hostname: server2.example.com
Uuid: 00457899-907c-485c-a1c1-949ff3b2999b
State: Peer in Cluster (Connected)
root@ubuntu:~#
```

Şekil.60. Sunucuların depolamadaki durumu.

2 kopyalı depolama alanı oluşturalım, bu komutu sadece bir sunucuda çalıştırıldığında depolama kopyası oluşmuş olur.

```
root@node1:~#gluster volume create yes replica 2 server1.example.com:/y
server2.example.com:/y
```

veya

```
root@node1:~#gluster volume create yes replica 2 192.168.10.156:/y
192.168.10.195:/y
```

```
root@ubuntu:~# gluster volume create yes replica 2 192.168.10.156:/y 192.168.10.195:/y
Creation of volume testvol has been successful. Please start the volume to access data.
```

Şekil.61. Depolama oluşturulması ve kopya sayısı.

Şimdi depolamayı başlatmak gerekiyor.

```
root@ubuntu:~# gluster volume start yes
volume start: yes: Starting volume testvol has been successful
root@ubuntu:~#
```

Şekil.62. Depolamanın başlatılması.

Depolamanın oluşturulduğunu göstermek gerekirse,

```
root@ubuntu:~# gluster volume info yes

Volume Name: yes
Type: Replicate
Volume ID: 72f899a9-fb68-4eef-a929-0330f9dc7bd9
Status: Started
Number of Bricks: 1 x 2 = 2
Transport-type: tcp
Bricks:
Brick1: server2.example.com:/y
Brick2: 192.168.10.156:/y
root@ubuntu:~#
```

Şekil.63. Depolama bilgilerinin kontrol edilmesi.

Oluşturulan sunucuların nerede yaratıldıkları `--instances_path` bayrağıyla belirtilmektedir. Yapılması gereken sunucuların oluşturulduğu kaynağın bu depolamaya bağlanmasıdır. Bunun için ilk önce varsayılan sunucuların depolandığı yeri bulmak gereklidir. Bulut sisteminde varsayılan `--instances_path` yolu ise `/var/lib/nova/instances`.

```
root@node1:~#mount -t glusterfs
server1.example.com:/yes/var/lib/nova/instances
root@node2:~#mount -t glusterfs server2.example.com:/yes
/var/lib/nova/instances
```

Sunucudaki disk kullanımını görüntüleyelim.

```

root@ubuntu:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda2       40G   12G   26G   32% /
udev            1.5G   12K   1.5G    1% /dev
tmpfs           607M   936K   606M    1% /run
none            5.0M    0    5.0M    0% /run/lock
none            1.5G  128K   1.5G    1% /run/shm
cggroup         1.5G    0    1.5G    0% /sys/fs/cgroup
server1.example.com:/yes 40G   14G   24G   38% /var/lib/nova/instances

```

Şekil.64. Sunucu disk kullanım durumu.

Bir test dosyası ile yeni oluşturulan yüksek erişilebilirlikli depolamayı test edelim. İlk önce node1 sunucusunda “selim.test” isimli bir klasör oluşturuldu.

```

root@ubuntu:/var/lib/nova/instances# mkdir selim.test
root@ubuntu:/var/lib/nova/instances# ls -l
total 16
drwxrwxrwx 2 nova nova 4096 Dec 17 15:00 base
drwxrwxr-x 2 nova nova 4096 Dec 17 15:48 instance-000000e0
drwxrwxrwx 2 nova nova 4096 Nov  7 17:06 locks
drwxr-xr-x 2 root root 4096 Dec 28 21:25 selim.test
root@ubuntu:/var/lib/nova/instances#

```

Şekil.65. Depolama test işlemi.

Aynı klasörün saniyeler içerisinde node2 sunucusunda oluşturulan çevik depolama alanına kopyalandığını görebiliriz.

```

root@ubuntunode2:/y# ls -l
total 24
drwxrwxrwx 2 nova nova 4096 Dec 17 15:00 base
drwxrwxr-x 2 nova nova 4096 Dec 17 15:48 instance-000000e0
drwxrwxrwx 2 nova nova 4096 Nov  7 17:06 locks
drwxr-xr-x 2 root root 4096 Dec 28 21:25 selim.test
root@ubuntunode2:/y#

```

Şekil.66. Depolama test sonucu.

4.1.3. Canlı Taşıma Yapılandırması

Sanal makine disklerinde replikasyonlu depolama kullanıldığından dolayı canlı taşıma işlemi blok canlı taşıma işleminden kat kat hızlı olacaktır. Blok taşıma işleminde işlemci ve ağ gücü kullanıldığından bu işlem esnasında performans kaybı yaşanırken replikasyonlu sistemlerde performans kaybı yaşanmamaktadır.

4.1.3.1. Libvirt ve Nova Yapılandırması

libvirtd.conf dosyasını düzenleyelim.

```
root@ubuntu:~# nano /etc/libvirt/libvirtd.conf
```

Şekil.67. Libvirt konfigürasyonu düzenleme komutu.

```
öncesi : #listen_tls = 0
sonrası: listen_tls = 0
öncesi : #listen_tcp = 1
sonrası : listen_tcp = 1

yeni eklenen: auth_tcp = "none"
```

libvirt-bin.conf dosyasını düzenleyelim.

```
root@ubuntu:~# /etc/init/libvirt-bin.conf
```

Şekil.68. Libvirt-bin konfigürasyonu düzenleme komutu.

```
öncesi : exec /usr/sbin/libvirtd -d
sonrası : exec /usr/sbin/libvirtd -d -l
```

Diğer libvirt-bin dosyasını düzenleyelim.

```
root@ubuntu:~# nano /etc/default/libvirt-bin
```

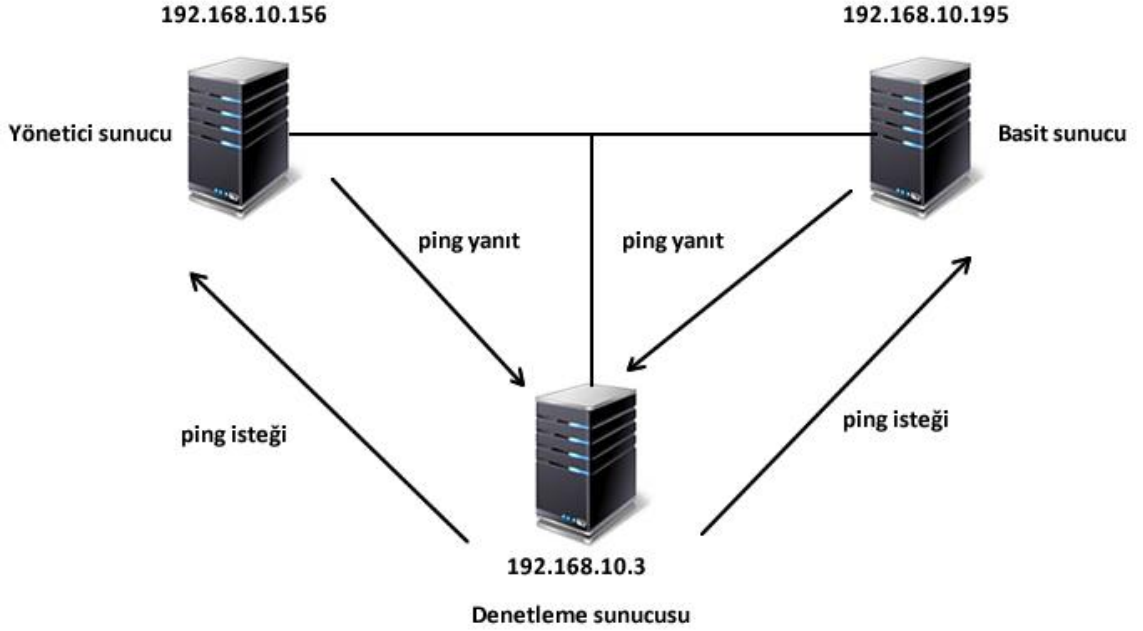
Şekil.69. Diğer libvirt-bin konfigürasyonu düzenleme komutu.

```
before :libvirtd_opts=" -d"
after  :libvirtd_opts=" -d -l"
```

libvirt servisini yeniden başlatalım.

```
root@ubuntu:~# service libvirtd restart
```

Şekil.70. Libvirt servisi yeniden başlatma komutu.



Şekil.71. Sistemin kontrolü.

Sistemin çalışır durumda olup olmadığını denetleme sunucusu ile kontrol edebiliriz. Bu sunucu yönetici ve basit sunucuya belli aralıklarla ping isteğinde bulunur ve bu sunucudan yanıt bekler. Eğer hedefteki sunucu bu isteğe cevap verirse sistemin çalışır durumda olduğunu anlamaktayız. Ancak ping isteğine yanıt verilmemesi durumunda diğer çalışır durumdaki sunucuya erişilemeyen sunucudaki yükün taşınması gerektiği isteğini göndeririz. Böylece taşıma işlemine başlanmış olur. Bu sistemde canlı taşıma yöntemi kullanıldığından ve aslında cevap vermeyen sunucu üzerindeki yük diğer sunucularda replike edildiğinden dolayı bu işlem hızlıca tamamlanacaktır. Sistemi kullanan kullanıcıların sunucuları anlık olarak kesinti yaşayıp hemen akabinde sunucuları aktif hale gelecektir.

```

#!/usr/bin/env python
#Bu script sistemdeki fiziki makinelerin donanımsal veya yazılımsal olarak arızalandığında sanal makinelerin kurtarılması için
#olusturulmuştur.
# - Veri tabanında host bilgilerini duzenler
# - ilgili fiziki makinede sanal sunucuyu spawn moduna getirir
#
#Kullanımı: ./nova-compute-recovery -i INSTANCE_ID [-f HOSTNAME]

import gettext
import os
import sys
from optparse import OptionParser
from socket import gethostname

gettext.install('nova', unicode=1)

from nova import context
from nova import db
from nova import flags
from nova import utils
from nova.network import manager
from nova.volume import manager
from nova.virt.connection import get_connection
from nova.virt.libvirt import connection as libvirt_conn
FLAGS = flags.FLAGS

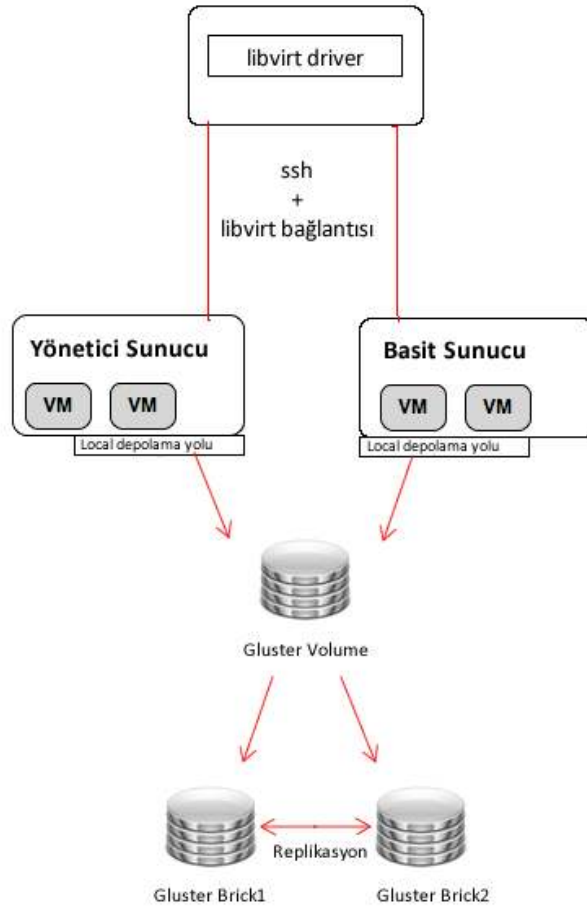
def main():
    instance_id = sys.argv[2]
    print "instance_id:", instance_id
    recovery_host = sys.argv[3]
    recovery_host = recovery_host.replace("]", "")
    print "recovery_host:", recovery_host
    FLAGS.connection_type = "libvirt"
    ctxt = context.get_admin_context()
    libvirt = get_connection(read_only=False)
    instance_ref = None
    try:
        instance_ref = db.instance_get(ctxt, instance_id)
        db.instance_update(ctxt, instance_ref.id, {'host': recovery_host})
    except Exception, e:
        print "Exception: ", e
    try:
        virt_dom = libvirt.lookup_by_name(instance_ref['name'])
    except:
        virt_dom = None

    if virt_dom:
        try:
            virt_dom.undefine()
        except:
            raise Exception('Could not undefine domain: %s' % virt_dom)
    libvirt.spawn(instance_ref, libvirt_conn.get_network_info(instance_ref))
    info = libvirt.get_info(instance_ref['name'])
    db.instance_set_state(ctxt, instance_ref.id, info['state'])

if __name__ == '__main__':
    main()

```

Şekil.72. Aktif olmayan bulut sisteminden sunucu kurtarma işlemi.



Şekil.73. Bulut sistemi ve depolama çalışma şeması.

Bu sistemde standart kurulum ile libvirt sürücü dosyasında kodsız değişiklik yapmak gerekmektedir. Aynı şekilde standart yükleme ile gelen nova konfigürasyon dosyasında yeni bir bayrak eklememiz gerekmektedir.

Nova konfigürasyon dosyasına canlı taşıma için yeni bir bayrak ekleyelim.

```
live_migration_flag=VIR_MIGRATE_UNDEFINE_SOURCE, VIR_MIGRATE_PEER2PEER, VIR_MIGRATE_LIVE, VIR_MIGRATE_UNSAFE
```

Şekil.74. Nova canlı taşıma satırı.

Bu işlemler ile birlikte artık bulut sisteminde;

- Hizmet kesintisi olmadan sanal sunucuların canlı şekilde taşınması,
- Farklı hipervizör üzerinde olsa dahi sanal sunucuların canlı şekilde devamlılığı,
- Yük devretme ve yedeklilik sistemi ile daha çevik bir yapıya kavuşmuş olur.

4.1.4. Sistemin Test Edilmesi

Sistemde oluşturduğumuz sunucuların listesi görüntülemek için:

```
root@ubuntu:~# nova list
```

Şekil.75. Sanal sunucuları listeleme komutu.

Sistemdeki tüm sanal sunucular aşağıdaki gibi görüntülenecektir.

ID	Name	Status	Networks
134c4a8b-7cac-47ca-8b1d-37ffecf26778	cirros	HARD_REBOOT	vlan1=10.10.10.3
69eb9ff9-11f7-4fbc-b913-1857dc58f74d	mini	ACTIVE	vlan1=10.10.10.4

Şekil.76. Sistemdeki sanal sunucular.

Canlı taşıma işlemini gerçekleştireceğimiz sunucunun detaylarını görüntülemek için:

```
root@ubuntu:~# nova show 69eb9ff9-11f7-4fbc-b913-1857dc58f74d
```

Şekil.77. Sanal sunucunun detaylarını görüntüleme komutu.

Sunucu detayları aşağıdaki gibi görünecektir.

Property	Value
OS-DCF:diskConfig	MANUAL
OS-EXT-SRV-ATTR:host	ubuntu
OS-EXT-SRV-ATTR:hypervisor_hostname	server1.example.com
OS-EXT-SRV-ATTR:instance_name	instance-000000e1
OS-EXT-STS:power_state	1
OS-EXT-STS:task_state	None
OS-EXT-STS:vm_state	active
accessIPv4	
accessIPv6	
config_drive	
created	2013-12-17T13:54:04Z
flavor	small (22)
hostId	48202fbc39d06d7753a564dfcc9ff131f0fc7d4ef3a805db7e2c1cca
id	69ebeff9-11f7-4fbe-b913-1857dc58f74d
image	mini 11.10 (82415b56-3b21-4a6f-ae4b-ebc56a426d44)
key_name	None
metadata	{}
name	mini
progress	0
security_groups	[[{'name': 'u'default'}]]
status	ACTIVE
tenant_id	0b6df7b5ac524fda8f0c548e5fab8e4b
updated	2014-01-13T19:21:25Z
user_id	377f22bab5454df0be9e059686f89dd2
vlan1 network	10.10.10.4

Şekil.78. Sanal sunucu detayları.

Diğer fiziksel makineye canlı taşıma işlemini gerçekleştirelim.

```
root@ubuntu:~# nova live-migration mini ubuntu2node2
```

Şekil.79. Canlı taşıma komutu.

Property	Value
OS-DCF:diskConfig	MANUAL
OS-EXT-SRV-ATTR:host	ubuntu2node2
OS-EXT-SRV-ATTR:hypervisor_hostname	server2.example.com
OS-EXT-SRV-ATTR:instance_name	instance-000000e1
OS-EXT-STS:power_state	1
OS-EXT-STS:task_state	None
OS-EXT-STS:vm_state	active

Şekil.80. Canlı taşıma işlemi sonucu.

5. SONUÇ

Bulut kullanımına dönüşüm başlamıştır ve hızlanarak devam edecektir. Standartların geliştirilmesi ve sistemin yaygınlaştırılması ile birlikte günlük yaşantıda bulut kelimesi ile daha fazla karşılaşacağımız gibi görünmekte. Teknolojik cihazların hayatımıza girmesiyle birlikte bulut kullanımında da artışlar meydana gelmektedir. Akıllı telefon satın alındığında harici bulut depolama hizmetleri verilmeye başlandı, diğer bir örnek ise Google firmasının çıkarttığı Google Glass ürünü sadece bulut sistem gücünün bir küçük yansıması gibi.

Bulut bilişim bize uygun maliyetli, bakım gerektirmeyen, ölçeklenebilir ve belki de en önemlisi BT altyapısına yüksek yatırımlar yapmayıp, istediğimiz zaman vazgeçebileceğimiz hizmetler sunuyor. İşletmeler bakım ve onarım gibi alt yapı işletimi ile uğraşmak yerine, kendi işlerine odaklanabiliyor.

Bulut bilişimi kullanmaktan çekinme sebeplerinin başında güvenlik konusu gelmektedir. Bu konuda gelinen durumu bir örnekle açıklamak gerekirse vahşi batı buna örnek verilebilir. Eski tarihleri aklımızda getirelim, kovboy kasabasında bir banka olduğunu hayal edelim. Bankaya paralar yatırılır ancak ertesi gün bu bankalar soyulur. Bu yerde parayı sahibinin saklaması bankaya koymaktan daha güvenliydi. Ancak yıllar içerisinde bankalar güçlendi, daha iyi korunur oldu. Bulut olayı bir zaman, süreç ve günlük hayatımızda yer alma durumudur. İlk zamanlarda insanlar tereddüt yaşar ancak zamanla verilerin kendi altyapısında tutulması riskli hale geldiğinde daha rahat şekilde buluta geçiş olacaktır. Hem teknoloji hem bilgi güvenliği hızla gelişmektedir.

Bulut bilişim altyapı sağlayıcısı firmaların en büyük sıkıntılarından biri hizmetlerinde kesinti yaşanmasıdır. Bu kesintiler ile müşteriler maddi kayıplar yaşayabilmektedir. Bu neden yedeklilik büyük bir önem taşımaktadır. Yedekli sistem kurulumları maliyetleri yukarıya çıkarsa da sağladıkları açısından çok önemli problemleri ortadan kaldırmaktadır. Bu tez çalışmamda yüksek erişilebilir bir bulut mimarisi oluşturmayı hedeflenmiştir.

KAYNAKLAR

- [1] <http://www.computing.co.uk/ctg/news/2141622/amazon-enterprise-cloud-backup-provider>, January 26, 2012.
- [2] Stevens, A. , "When hybrid clouds are a mixed blessing", The Register, Retrieved March 28, 2012.
- [3] Hamdaqa, M. , "A Reference Model for Developing Cloud Applications", International Conference on Cloud Computing and Services Science, 2011.
- [4] Uğurlu, S. , "Bulut Bilişim", <http://www.suatugurlu.com/?p=120>, Temmuz, 2012.
- [5] Özyücel, N. , "Windows Azure Türkiye’de", <http://blog.microsoft.com.tr/windows-azure-turkiyede.html>, Haziran, 2013.
- [6] Yüksel, H. , "Bulut Bilişim El Kitabı", <http://yuksehis.files.wordpress.com/2012/01/bulutbilic59fimekitabc4b1.pdf>, Ocak, 2012.
- [7] Dinçkan, A., İŞ SÜREKLİLİĞİ YÖNETİM SİSTEMİ KURULUMU, 20 Ekim, 2008.
- [8] Güner, K., <http://kadirguner.com/linux/glusterfs-ile-replication>, 13 Kasım, 2013.

ÖZGEÇMİŞ

10 Haziran 1985 tarihi, Rize İli Kalkandere ilçesi doğumluyum. İlk, Orta ve Liseyi İstanbul İli Üsküdar ilçesinde tamamladıktan sonra, 2004 yılında Girne Amerikan Üniversitesi, Mimarlıkve Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölüme kayıt oldum. Bu bölümden 2009 yılında mezun olduktan sonra 2010 yılında özel bir şirkette yazılım bölümünde çalışmaya başladım. 2011 yılında, Beykent Üniversitesi Bilgisayar Mühendisliği Anabilim Dalında yüksek lisans eğitimine başladım.

Özel ilgi alanlarım, son teknolojik gelişmeleri takip etmek, mobil uygulama geliştirmektir.

Yabancı dil olarak İngilizce bilmekteyim.

Aday: Selim AKSOY