



WEBLAB OPTİMİZASYON

Ömer KURT

**Haziran 2020
DENİZLİ**

WEBLAB OPTİMİZASYON PROJESİ

Pamukkale Üniversitesi
İktisadi ve İdari Bilimler Fakültesi
Yönetim Bilişim Sistemleri Bölümü
Lisans Tezi

Ömer KURT

Danışmanlar:
Prof. Dr. Selçuk Burak HAŞILOĞLU

Haziran 2020

DENİZLİ

ÖZET

WEBLAB OPTİMİZASYON PROJESİ

ÖMER KURT

Lisans Tezi

Türkiye

Yönetim Bilişim Sistemleri Bölümü

Tez Yöneticisi: Prof. Dr. Selçuk Burak HAŞILOĞLU

Haziran, 2020

Bilgisayar teknolojisinin gelişimi birçok sorunu ortadan kaldırdığı gibi beraberinde yeni sorunlar ortaya çıkarmıştır. Bilgisayar kullanıcılarının en çok karşılaştığı problemler arasında, çok fazla uygulamanın yüklü olduğu bilgisayarların yavaşlaması, bazı uygulama gereksinimlerinin bilgisayar donanım özellikleri tarafından karşılanamaması, buna bağlı olarak bilgisayar/donanım yenileme maliyetlerinin ortaya çıkması ve kapalı kaynak kodlu yazılımların lisans maliyetleri gösterilebilir.

Weblab adını verdiğimiz projede, bu problemlere odaklanılmaktadır. Projemizi gerçekleştirmek adına bölümümüz laboratuvarını pilot bölge olarak seçmiş bulunmaktayız. Weblab projemiz, bölümümüz laboratuvarında yaşanan şu problemlere de çözüm üretmektedir:

- Laboratuvara yeni bir uygulama yüklemek gerektiğinde tüm bilgisayarlara tek tek yüklemek.
- Bir bilgisayarda çok fazla uygulama yüklü olduğu için bilgisayarlarda yaşanan yavaşlama.
- Bazı uygulamaların gereksinimleri, bilgisayar donanım özelliklerinin üstünde olması.
- Uygulama gereksinimlerinden doğan bilgisayar/donanım yenileme maliyetleri.
- Kapalı kaynak kodlu yazılımların lisans maliyetleri.

Anahtar Kelimeler: Konteyner, Docker, Pardus, Weblab, VNC

İÇİNDEKİLER

ÖZET.....	3
GİRİŞ.....	5
1. Projenin Amacı ve Önemi.....	5
1.1. Projenin İçerdiği Yenilik (Özgünlük) Unsuru.....	6
1.2 Weblab Benzeri Sistem ve Uygulamalar.....	6
2. Kullanılan Yöntem ve Teknikler.....	7
3. Docker Konfigürasyonu.....	10
3.2. Sanal Makine / Weblab Ekranı.....	11
SONUÇ VE TARTIŞMA.....	12
KAYNAKÇA.....	13

GİRİŞ

Teknoloji her geçen gün değişmekte ve bu değişmekte olan teknolojiye ayak uydurmak zorunlu hale gelmektedir. Değişim sadece donanımlarda değil, yazılım sektöründe de hızla gerçekleşmektedir. Bu hızlı büyümenin doğurduğu problemler arasında donanımsal maliyetlerdeki artış ve yazılımsal uygulamaların boyutları, kullanıcılar açısından büyük bir problem haline gelmiştir. Kullanıcılar için daha fazla alan ve daha güçlü donanım ihtiyacı doğmuştur. Kullanıcılar bu ihtiyacı yeterli derecede karşılayamadığı takdirde işlerinde yavaşlama ve buna bağlı olarak zaman kaybı yaşamaktadır.

Mevcut donanımlar, gelişen teknoloji karşısında hızla eskimekte veya yetersiz kalmaktadır. Bu durum, üniversiteler ve firmalar başta olmak üzere, bilgisayar merkezli çalışan tüm kurumlara mali açıdan zarar vermektedir. Üniversiteler ve şirketler de bu durumdan olumsuz etkilenmekte, yıllar geçtikçe bilgisayar laboratuvarları donanımsal gereksinimlerini karşılamakta geri kalmaktadır. Hızlı yazılımsal gelişim, donanımları kısa sürede üst modele yükseltmeye zorlamaktadır. Bu durum, donanım iyileştirme maliyetlerini sürekli hale getirmektedir.

Üniversite laboratuvarlarından yola çıkarak yaptığımız gözlemler sonucu, çok sayıda bilgisayarların bir arada bulunduğu ortamlarda, yüklenmesi gereken bir uygulamayı tüm bilgisayarlara tek tek yüklemek zorunda kalıyoruz. Bu durum haliyle zaman kaybı yaşatmakta ve işlevsel olmamaktadır. Ayrıca, bir laboratuvar da sorunsuz çalışan uygulama, diğer bir laboratuvardaki bilgisayarlarda sorun çıkarabilmektedir.

Weblab adlı projemizde, uygulamaları web ortamında açarak, donanım yenileme masraflarını büyük ölçüde ortadan kaldırmak, donanıma bağlı yavaşlamaların önüne geçmekle yerel bilgisayarımıza ekstra bir yükleme işlemi yapmadan, gerekli uygulamalara kolayca erişilebilmeyi sağlamaktayız. Ayrıca milli işletim sistemi dağıtımımız olan Pardus'u kullanarak, kapalı sistemlere olan bağımlılığımızı azaltmayı hedefliyoruz.

1. Projenin Amacı ve Önemi

Weblab adlı projemizde, üniversitelerdeki ve şirketlerdeki bilgisayarların donanımlarını her zaman yenilemektense, herhangi bir web tarayıcısı üzerinden belirlenen web adresine girerek giriş yaptıktan sonra sistemde var olan uygulamalara erişebilmektedir. Bu uygulamalara web ortamı üzerinden erişilebileceği için kullanıcının bilgisayarı gelişmiş donanım özelliklerine sahip olmak zorunda değildir.

Weblab projesi ile kullanıcıların bilgisayarlarındaki donanıma bakılmaksızın, web tarayıcısını çalıştırabilen her bilgisayar, Weblab sistemine kayıtlı olan uygulamalara erişebilecektir.

Weblab projemizin ana faydaları, üniversitelerin bilgisayar laboratuvarlarını ve şirketlerin bilgisayarlarını daha etkin ve verimli kılabilcek olması, donanım yenileme maliyetlerini büyük ölçüde ortadan kaldırabilmesi, donanım kaynaklı yavaşlamalara engel olabilmesi ve her kullanıcıya, her yerden ulaşabileceği sanal bir makine vermesidir. Büyük teknoloji devlerinin, son yıllarda üzerine düştüğü sanallaştırma teknolojilerine gerekli farkındalığı ve önemi yaratmaya çalışmaktayız. GNU/Linux işletim sistemi tabanlı milli dağıtımımız olan Pardus'u projemizin parçası haline getirerek hem yaygınlaştırmayı hem de özgür yazılım felsefesini aşlamayı hedefliyoruz. Bu hedefimizdeki temel amacımız, pahalı lisanslara sahip olan işletim sistemlerine bağımlılığımızı azaltmaktır.

Türkiye'de 129'u devlete ait olmak üzere toplamda 207 adet üniversite bulunmaktadır (yök, 2019). Buradan bir çıkarım yapılacak olursa Weblab projesi bu 207 üniversitede bulunan, sayısı toplamda yüzleri ve belki bini bulabilecek bilgisayar laboratuvarlarının tamamının sorunlarını ortadan kaldırabilecektir.

1.1. Projenin İçerdiği Yenilik (Özgünlük) Unsuru

Weblab projemizde, ülkemizdeki üniversite ve şirketleri ele aldığımız için, projemiz ulusal düzeyde yenilik kategorisine girmektedir. Weblab projemiz, geleneksel bilgisayar kullanım alışkanlığını değiştirmektedir. Weblab projesinin kattığı yenilikler şunlardır:

- Docker Container teknolojisini kullanarak birbirinden izole edilmiş uygulamalar kullanmak.
- Projemizin sitesine girildiğinde UX (kullanıcı deneyimi) esas alınarak yeni bir kullanıcı arayüzü inşa etmek.
- Üniversite laboratuvarlarındaki geleneksel düzeni, modern ve verimli hale getirmek.

Weblab projemizi oluşturan uygulamalar GNU, Apache License Version 2.0 ve Creative Commons Attribution 3.0 Unported gibi açık kaynaklı lisanslara sahip olduğundan ötürü, projemiz, ticari olarak kullanılabilir, değiştirilebilir ve dağıtılabilir.

1.2 Weblab Benzeri Sistem ve Uygulamalar

Weblab projemize benzer sanal laboratuvarları olan üniversiteler şunlardır:

- Calgary Üniversitesi
- Arizona Üniversitesi
- Wisconsin-Madison Üniversitesi
- Akron Üniversitesi

Benzer projeleri inceleyip gerekli deneme çalışmalarını yapmamızın ardından, Weblab projemiz diğer uygulamalara kıyasla şu avantajları içermektedir:

- Kullanılabilirlik kapsamı diğer uygulamalara göre oldukça geniştir.
- Eğitim için kullanılmaya elverişlidir.
- Güvenlik ve devamlılık açısından oldukça etkilidir.
- Sistemin altyapısı ve kontrolüne hakim olmak mümkündür.

Ayrıca Weblab projemizin sistemine benzer bir sistem olan Akron Üniversitesi'nin Virtual Lab projesi ile Weblab projimizi kıyaslayaması Tablo 1'de verilmiştir

Tablo 1. Weblab ve Virtual Lab karşılaştırması.

	Weblab	Virtual Lab
Konteyner teknolojisi kullanımı	✓	X
GNU Linux işletim sistemi	✓	X
Çekirdek seviyesinde sanallaştırma	✓	X
Özelleştirilebilir	✓	X
Kullanıcı verilerini saklı tutabilme	✓	✓
Her yerden erişilebilir	✓	✓
Açık kaynaklı uygulamaların ön yüklü olarak gelmesi	✓	X
Açık kaynak koda sahip olma	✓	X
Düşük maliyet	✓	X
Uzun süreli (LTS) destek	✓	X

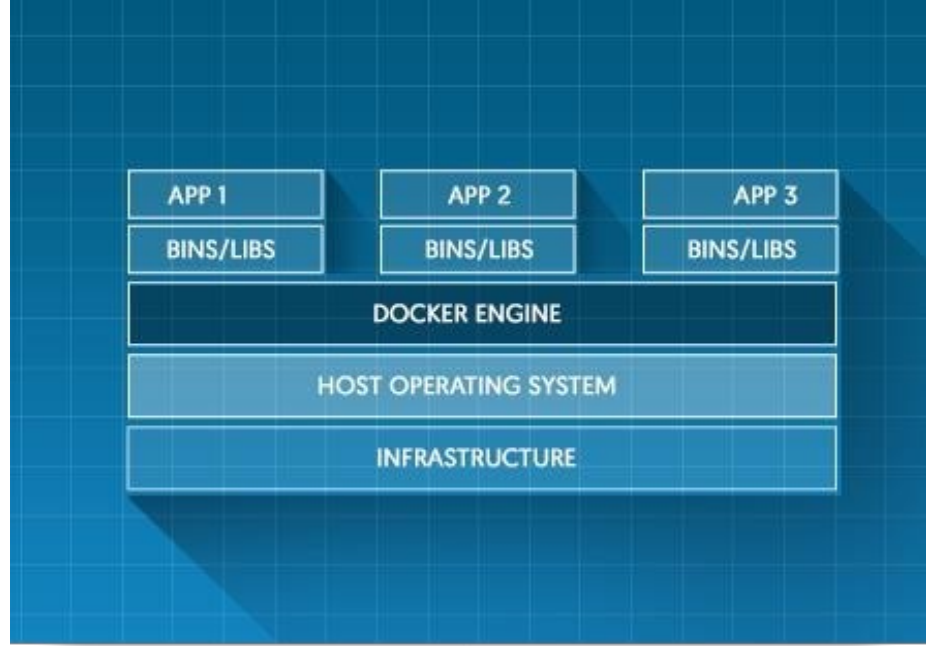
2. Kullanılan Yöntem ve Teknikler

Weblab projemiz, tarayıcı üzerinden erişilebilecek sanal makinelerdir. Bu makineler, yerel bilgisayarımızda ne yapabiliyorsak hatta yapamadıklarımızı bile yapmamıza imkan tanır. Bu projemizi oluşturan teknolojiler ve yazılımlar şunlardır:

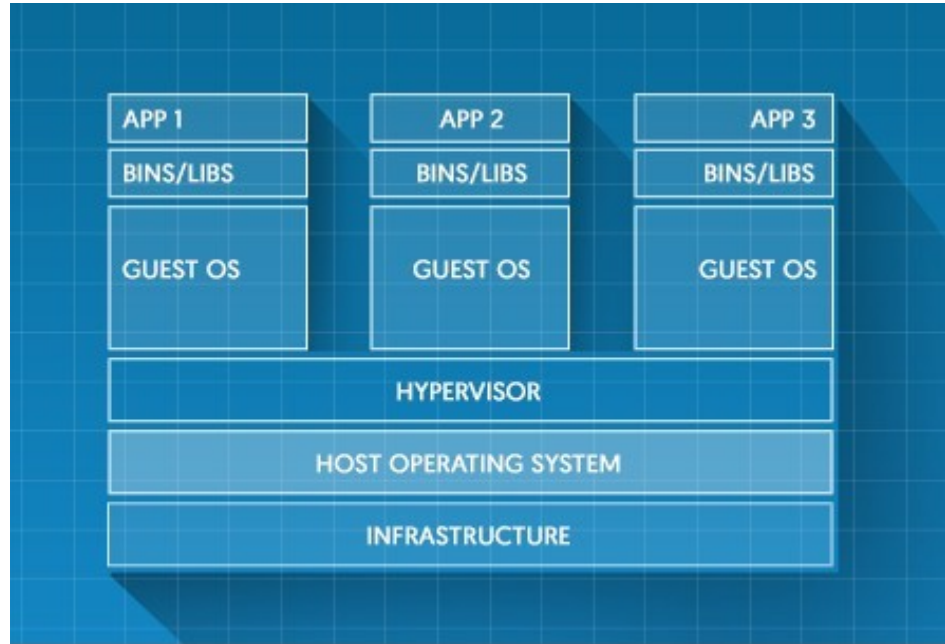
- Docker
 - Container
 - Docker Image
- Portainer.io
- TightVNC Server / TigerVNC Server

- Pardus / Fedora
- Apache Guacamole / noVNC

Weblab projemizde, Docker kullanmamızın en büyük nedenlerinden bir tanesi konteyner teknolojisi ile Hypervisor teknolojisi arasındaki farktır. Bu farkın en büyük belirtisi disk alanında önemli bir tasarruf sağlamasıdır.



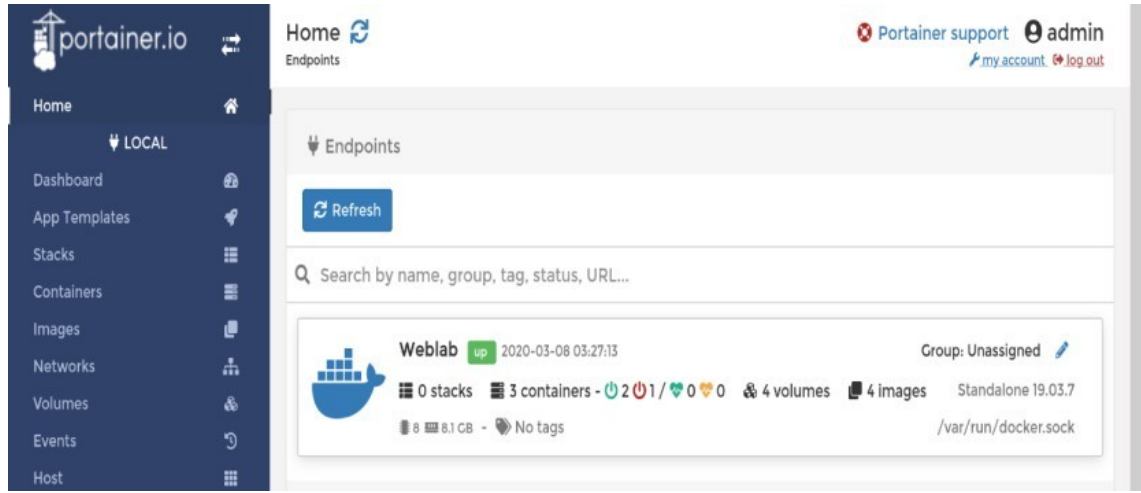
Şekil 1. Docker Engine çalışma mantığı



Şekil 2. Virtual Machine çalışma mantığı

Sanal işletim sistemlerinin her birine 7 GB disk alanı, 1 Core ve 2 GB RAM verdiğimiz düşünelim. Bu sanal işletim sistemlerinden 20 tane olduğunu varsayarsak, toplam $20 \times 7 = 140$ GB disk alanı, $20 \times 1 = 20$ Core ve $20 \times 2 = 40$ GB RAM gerekmektedir. Kullanıcı sistemi kullanmıyor olsa dahi sistem bunları sabit olarak kullanır. Şekil 2’de bu çalışma mantığı anlatılmaktadır. 20 adet sanal sunucu kurmak yerine Şekil 1’de gösterilen Docker Engine ile sanallaştırma yaptığımızda sadece 7GB disk alanı, 2GB RAM ve 1 Core kullanmak yeterli olacaktır. Bu da altyapıdan (infrastructure) büyük bir iş yükünü ortadan kaldıracaktır.

Portainer, Docker ortamlarını terminalle çok az uğraşarak hatta hiç uğraşmayarak Docker ile alakalı yönetim işlerinin Şekil 3’de görüldüğü gibi yapabilmemizi sağlar. Portainer, ayrıca Docker Konteynerları oluşturabilmemizi ve Docker üzerinde gözlemlerimizi web arayüzü ile uç birim ihtiyacımız olmadan yapabilmemize olanak sağlar. (Yıldırım, 2019)



Şekil 3. Portainer ile projemizi yönetme ekranı

VNC, özetle herhangi bir ağ sunucusu üzerinde çalışan grafik arayüz uygulamalarına herhangi başka ağ üzerindeki bir bilgisayardan kolaylıkla ve hızla erişerek bu uygulamaları kullanabilmemizi ve yönetmemizi sağlayan platform, bağımsız bir yapıdır (tigerVNC, 2019).

Apache Guacamole, istemcisiz bir uzak masaüstü ağ geçididir. Herhangi bir eklenti veya istemci yazılımı gerektmediği için bunu istemcisiz olarak adlandırıyoruz. HTML5 sayesinde, Apache Guacamole bir sunucuya kurulduktan sonra, masaüstlerinize erişmek için tek ihtiyacınız olan bir web tarayıcısıdır (guacamole, 2020).

noVNC, açık kaynaklı VNC istemcisidir. Hem bir VNC istemcisi hem de JavaScript kütüphanesinin üzerine kurulmuş bir uygulamadır. noVNC, mobil tarayıcılar (iOS ve Android) dahil olmak üzere herhangi bir modern tarayıcıda verimli çalışmaktadır. Projemizde, Apache Guacamole’nin alternatifi olarak yer almaktadır (novnc, 2019).

3. Docker Konfigürasyonu

```
systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; vendor prese
   Active: active (running) since Sat 2020-03-07 23:17:12 +03; 3h 53min ago
     Docs: https://docs.docker.com
    Main PID: 13029 (dockerd)
      Tasks: 46
     Memory: 63.3M
    CGroup: /system.slice/docker.service
           └─13029 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/contain
             13195 /usr/bin/docker-proxy -proto tcp -host-ip 0.0.0.0 -host-port 9
             13209 /usr/bin/docker-proxy -proto tcp -host-ip 0.0.0.0 -host-port 8
             20066 /usr/bin/docker-proxy -proto tcp -host-ip 0.0.0.0 -host-port 8

Mar 07 23:17:12 m13 dockerd[13029]: time="2020-03-07T23:17:12.133886681+03:00" le
Mar 07 23:17:12 m13 dockerd[13029]: time="2020-03-07T23:17:12.151148706+03:00" le
Mar 07 23:17:12 m13 systemd[1]: Started Docker Application Container Engine.
Mar 08 01:34:50 m13 dockerd[13029]: time="2020-03-08T01:34:50.591692932+03:00" le
Mar 08 01:34:50 m13 dockerd[13029]: time="2020-03-08T01:34:50.630830416+03:00" le
Mar 08 01:34:50 m13 dockerd[13029]: time="2020-03-08T01:34:50.630913364+03:00" le
Mar 08 01:37:27 m13 dockerd[13029]: time="2020-03-08T01:37:27.222874158+03:00" le
Mar 08 01:37:27 m13 dockerd[13029]: time="2020-03-08T01:37:27.259056541+03:00" le
Mar 08 01:37:27 m13 dockerd[13029]: time="2020-03-08T01:37:27.259145273+03:00" le
Mar 08 01:37:39 m13 dockerd[13029]: time="2020-03-08T01:37:39.766403240+03:00" le
lines 1-23/23 (END)
```

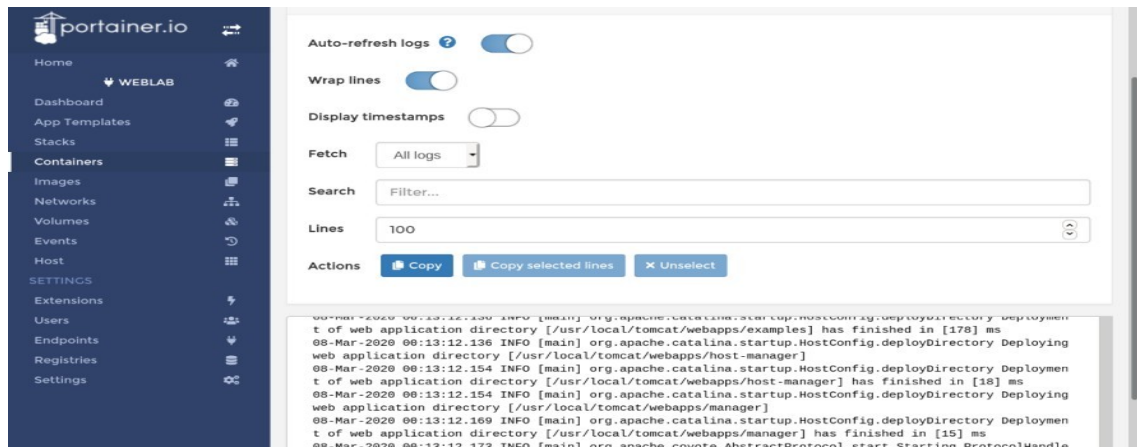
Şekil 4. Docker Engine'nin aktif durumu.

Weblab projemizi inşa etmek için ilk olarak server konfigürasyon ayarlarımızı yapıyoruz.

```
docker container ls
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        NAMES
89ab236a224c   oznu/guacamole:latest              "/init"                 9 days ago    Up 9 minutes   Guacamole
8c2646d1d99a   portainer/portainer                "/portainer"            9 days ago    Up 4 hours    portainer
```

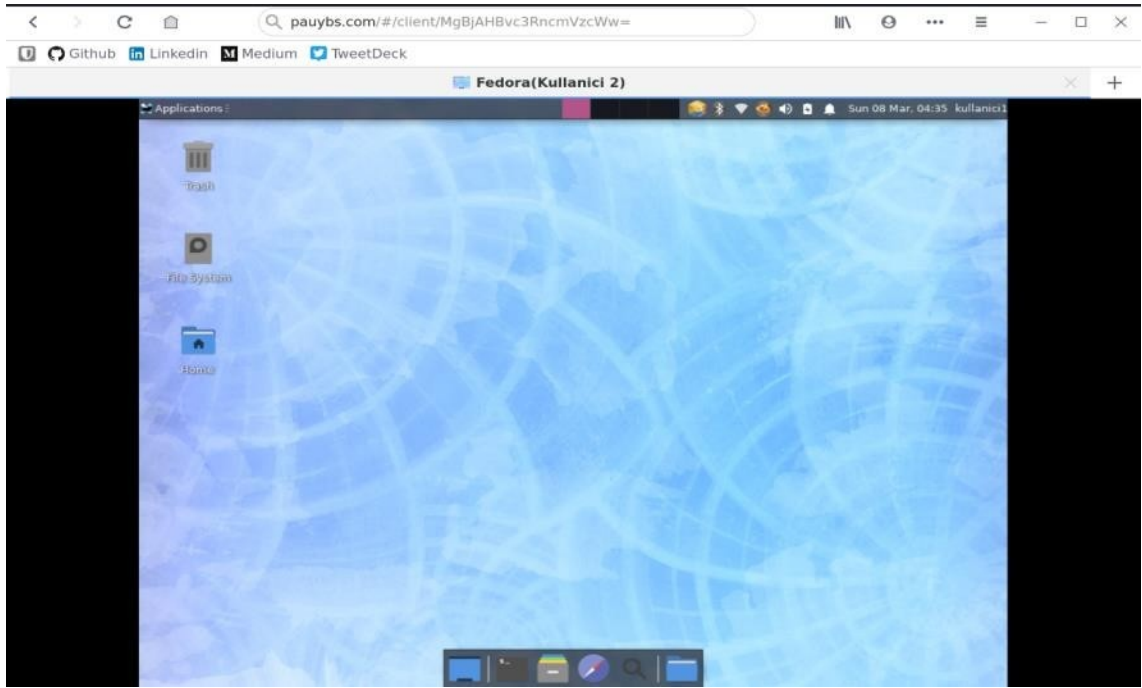
Şekil 5. Gerekli konteynerlerimizin listesi.

3.1. Portainer Konfigürasyonu



Şekil 6. Portainer Ekranı.

3.2. Sanal Makine / Weblab Ekranı



Şekil 7. Sanal Makine

Kullanıcıların, Weblab projesine ait web sitesine girdiklerinde karşılaşacakları sanal makine ekranı Şekil 7’de görüldüğü gibi olacaktır. Kullanıcı, bu ekran ile gerekli uygulamalara erişim sağlayacaktır.

SONUÇ VE TARTIŞMA

Weblab projesi, internet hızı ile doğrudan alakalıdır. Haliyle yaşanabilecek problemlerin büyük bir kısmı, server ile kullanıcı arasındaki bağlantı sorunları olacaktır. Projemizin verimli şekilde kullanılabilmesi için internet ağının ortalama 20 MB indirme hızına ihtiyacı vardır. İnternet teknolojisinin her geçen gün gelişmesine paralel olarak Weblab verimliliği de gelişimini sürdürecektir. 5G teknolojisinin de ülkemize gelişi ile, bağlantı problemlerini büyük ölçüde ortadan kalkacaktır.

Mali açıdan, projemizin temel giderleri server kiralama ve domain ücretleridir. Projemizi gerçekleştirirken kullanılan uygulamalar, özgür yazılım uygulamaları olduğu için yüksek bir maliyet doğurmayacaktır. Kullanıcı için donanım maliyetlerini ise büyük ölçüde düşürmektedir.

Weblab projesi, alpha ve test aşamasındadır. Projenin tamamen bitirilmesi için, araştırma çalışmaları sürmektedir.

KAYNAKÇA

Guacamole. (2020, 1 29). Apache Guacamole: guacamole.apache.org adresinden alındı

Novnc. (2019, 12 21). Github: <https://github.com/novnc/noVNC> adresinden alındı

TigerVNC. (2019). TigerVNC: www.tigervnc.org adresinden alındı

Yıldırım, G. (2019, 5 26). Portainer ile Docker - 0: Başlangıç.

YÖK. (2019). YÖK: istatistik.yok.gov.tr adresinden alındı