

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**AÇIK KAYNAK UZAKTAN EĞİTİM
YAZILIMLARINDA ALTYAPI OTOMASYONU VE
ÇIKTILARIN DEĞERLENDİRİLMESİ**

YÜKSEK LİSANS TEZİ

Osman Bedrettin YÜCE

Enstitü Anabilim Dalı

**: BİLİŞİM SİSTEMLERİ
MÜHENDİSLİĞİ**

Tez Danışmanı

: Prof. Dr. İsmail Hakkı CEDİMOĞLU

Nisan 2022

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**AÇIK KAYNAK UZAKTAN EĞİTİM
YAZILIMLARINDA ALTYAPI OTOMASYONU VE
ÇIKTILARIN DEĞERLENDİRİLMESİ**

YÜKSEK LİSANS TEZİ

Osman Bedrettin YÜCE

**Enstitü Anabilim Dalı : BİLİŞİM SİSTEMLERİ
MÜHENDİSLİĞİ**

**Bu tez 25/04/2022 tarihinde aşağıdaki jüri tarafından oybirliği / oyçokluğu ile
kabul edilmiştir.**

**Prof. Dr.
İsmail Hakkı CEDİMOĞLU
Jüri Başkanı**

**Doç. Dr.
İhsan Hakan SELVİ
Üye**

**Dr. Öğr. Üyesi
Ahmet KARACA
Üye**

BEYAN

Tez içindeki tüm verilerin akademik kurallar çerçevesinde tarafımdan elde edildiğini, görsel ve yazılı tüm bilgi ve sonuçların akademik ve etik kurallara uygun şekilde sunulduğunu, kullanılan verilerde herhangi bir tahrifat yapılmadığını, başkalarının eserlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduğunu, tezde yer alan verilerin bu üniversite veya başka bir üniversitede herhangi bir tez çalışmasında kullanılmadığını beyan ederim.

Osman Bedrettin Yüce

25/04/2022

TEŐEKKÜR

Yüksek lisans eğitimim boyunca değerli bilgi ve deneyimlerinden yararlandığım, her konuda bilgi ve desteğini almaktan çekinmediğim, araştırmanın planlanmasından yazılmasına kadar tüm aşamalarında yardımlarını esirgemeyen, teşvik eden, aynı titizlikte beni yönlendiren değerli danışman hocam Prof. Dr. İsmail Hakkı Cedimođlu'na teşekkürlerimi sunarım.

Laboratuvar olanakları konusunda anlayış ve yardımlarını esirgemeyen Sakarya Üniversitesi Uzaktan Eğitim Araştırma ve Uygulama Merkezi Müdürü Prof. Dr. Mehmet Barış Horzum ile Sakarya Üniversitesi Bilgisayar Araştırma ve Uygulama Merkezi Müdürü Prof. Dr. Ahmet Özmen ve bilgi ve deneyimlerinden yararlandığım sayın hocalarım Öğr. Gör. Baran Kaynak ile Öğr. Gör. Uğur Özbek'e teşekkür ederim.

İÇİNDEKİLER

TEŞEKKÜR.....	i
İÇİNDEKİLER	ii
SİMGELER VE KISALTMALAR LİSTESİ.....	v
ŞEKİLLER LİSTESİ	vi
TABLolar LİSTESİ	vii
ÖZET.....	viii
SUMMARY	ix
BÖLÜM 1.	
GİRİŞ	1
BÖLÜM 2.	
İDEAL UZAKTAN EĞİTİM SİSTEMİ	4
2.1. BigBlueButton	5
2.2. Öğrenme Yönetim Sistemi.....	6
2.3. Yük Dengeleyici	8
2.4. Yardımcı Araçlar	10
2.4.1. TURN ve STUN sunucu	10
2.4.2. NFS dosya paylaşımı	11
2.5. Sunucuların İzlenmesi.....	11
2.5.1. Zaman serisi veritabanı	12
2.5.2. Ajan yazılımlar	12
2.5.3. Verilerin görselleştirilmesi	13
2.5.4. Toplantı aktivite verileri.....	14
2.5.5. Toplantı kayıtlarının işlenmesi.....	15

BÖLÜM 3.

UZAKTAN EĞİTİM SİSTEMİNİN KURULUMU	17
3.1. Altyapı Tercihi.....	18
3.1.1. Bulut teknolojisi.....	19
3.1.1.1. Altyapı bulut hizmeti	19
3.1.1.2. Platform bulut hizmeti	20
3.1.1.3. Yazılım bulut hizmeti	20
3.2. Uygun Ortamın Hazırlanması	20
3.3. Açık Kaynak Kodlu Yazılım	21
3.4. Yazılımın Kurulumu.....	21
3.5. Yardımcı Araçların Kurulumu.....	22
3.5.1. Yük dengeleyici kurulumu	22
3.5.1.1. Redis veritabanı	23
3.5.1.2. Postgre veritabanı	23
3.5.2. TURN ve STUN sunucu kurulumu.....	23
3.6. İnce Ayarlamaların Yapılması	23
3.7. Harcanan Efor	24

BÖLÜM 4.

ÖNERİLEN YÖNTEM.....	26
4.1. Kullanılan Araçlar	26
4.1.1. Ansible	27
4.1.2. Terraform	27
4.1.3. BASH Kabuğu	27
4.2. Kurulum	28
4.2.1. Gereksinimlerin belirlenmesi.....	29
4.2.2. Farklı ortamlar için betiklerin hazırlanması.....	30
4.2.3. BASH betik aşaması	30
4.2.4. Terraform aşaması.....	31
4.2.5. Ansible aşaması.....	31
4.3. Yazılım Bakımı	32
4.3.1. Sertifikaların yenilenmesi	32

4.3.2. Biriken dosyaların temizlenmesi.....	33
4.3.3. Paket güncellemeleri.....	34
4.4. Zaman Kazanımı.....	34
BÖLÜM 5.	
SONUÇLAR.....	35
KAYNAKLAR.....	38
EKLER.....	40
ÖZGEÇMİŞ.....	55

SİMGELER VE KISALTMALAR LİSTESİ

API	: Application Programming Interface
BBB	: Big Blue Button
CSH	: C Shell
DNS	: Domain Name System
GB	: Giga Byte
GNU	: GNU's Not Unix
GUI	: Graphical User Interface
HTTP	: Hyper Text Transfer Protocol
JSON	: JavaScript Object Notation
KSH	: Korn Shell
NAT	: Network Address Translation
NFS	: Network File System
SH	: Shell
SQL	: Structured Query Language
STUN	: Session Traversal Utilities for NAT
TCP	: Transmission Control Protocol
TCSH	: TENEX C Shell
TURN	: Traversal Using Relays around NAT
UDP	: User Datagram Protocol
UZEP	: Uzaktan Eğitim Platformu
vCPU	: Virtual Central Process Unit
ÖYS	: Öğrenme Yönetim Sistemi

ŞEKİLLER LİSTESİ

Şekil 2.1. İdeal BigBlueButton Yapısı	5
Şekil 2.2. BigBlueButton kullanıcı arayüzü.....	6
Şekil 2.3. UZEP fonksiyonlarının şeması	8
Şekil 2.4. UZEP yük dengeleyici WEB arayüzü.....	9
Şekil 2.5. TURN ve STUN sunucuların çalışma mantığı.	10
Şekil 2.6. Grafana aracı ile verilerin görselleştirilmesi.....	14
Şekil 2.7. Kayıt medyalarının oluşturulması.....	16
Şekil 3.1. Mevcut yapı	18
Şekil 3.2. Bulut hizmet modelleri.	19
Şekil 3.3. BigBlueButton kurulum tetikleme komutu.....	22
Şekil 4.1. Kurulum süreci.	28
Şekil 4.3. Temizlik yapan betiğin görev zamanlayıcı içerisindeki dosya yolu.	33

TABLolar LİSTESİ

Tablo 4.1. BigBlueButton donanım gereksinimleri.	29
Tablo 5.1. Klasik kurulum ve kod olarak altyapı kurulum farkı.	36

ÖZET

Anahtar kelimeler: bigbluebutton, uzaktan eğitim, bulut, açık kaynak, kod olarak altyapı

Uzaktan eğitim amacıyla kullanılan konferans yazılımları kullanıcıya oldukça basit ve anlaşılır bir arayüz sunsa da arka planında oldukça karmaşık bir yapı yer almaktadır. Böylesine karmaşık yapının sürekliliğini sağlamak bir yana dursun, ihtiyaç duyulduğunda yeni bir ortama aktarılması ve aynı kararlılığı elde etmek için oldukça uzun mesai harcanması gerekmektedir. Bu çalışma, bir üniversitede bulunan uzaktan eğitim altyapısındaki yapının ve yazılımların mümkün olduğunca otomatik bir şekilde kurulmasının gerçekleştirilmesi, gerekli bakımların yapılması ve gerçekleşen problemlere karşı alınan aksiyonların mümkün olduğunca nasıl otomatik hale getirildiği araştırılacaktır. Bunların yanında bu yapıdan sürekli olarak elde edilen verilerin işlenerek karar mekanizmalarına desteğini gösterecektir.

Her geçen gün hızla ilerleyen bilgisayar yazılımları, bu yazılımların yönetilmesi, bakımı, güncellemeleri, çalışma ortamları gibi problemleri beraberinde getirmiştir. Problemlere getirilen çözümler her ne kadar yardımcı olsa da ihtiyaçlara yönelik olarak birçok açıdan yetersiz gelmektedir. Bilgisayar ve bilişim altyapılarında kullanılan bu kaynakların etkili biçimde yönetimi çok önemlidir. Çeşitli nedenlerden ötürü artabilecek kaynak ihtiyaçlarını hızlıca genişletip gerek görülmediğinde daraltılması bu kaynakların maliyetinin yönetiminde oldukça etkilidir. Altyapı maliyetlerinde esneklik önemli olduğu kadar işlemler tekrar ettikçe zahmetli bir hal almaktadır. İhtiyacı gidermek için çeşitli otomasyon araçları bulunsa dahi yapılandırmalar göz önünde bulundurulduğunda her zaman istenilen sonucu verememektedir. Böyle durumlarda yapılandırmanın ihtiyacını karşılayacak çözümlerin bir kereye mahsus oluşturulması gerekmektedir.

Elde edilen kazanımlarda neredeyse yüzlerce aşamadan oluşan kurulum ve bakım sürecinin oluşturulan betiklerle ve konfigürasyon araçlarının kullanımıyla birkaç adıma düşürülerek ciddi anlamda zaman tasarrufu sağlandığı açıklanmıştır.

INFRASTRUCTURE AUTOMATION AND EVALUATION OF OUTPUT IN OPEN SOURCE DISTANCE EDUCATION SOFTWARE

SUMMARY

Keywords: bigbluebutton, distance learning, cloud, open source, infra structure as a code

Although conference software used for distance education offers a very simple and understandable interface to the user, there is a very complex structure in the background. Let alone ensuring the continuity of such a complex structure, it is necessary to spend a long time in order to transfer it to a new environment when needed and to achieve the same stability. This study will explain how the structure and software in the distance education infrastructure in a university are installed as automatically as possible, the necessary maintenance is carried out, and the actions taken against the problems that occur are automated as much as possible. In addition, the data obtained from this structure will be processed and will show its support to the decision mechanisms.

Computer software, which is advancing rapidly day by day, has brought problems such as the management, maintenance, updates and working environments of these software. Although the solutions brought to the problems are helpful, they are insufficient in many respects for the needs. Effective management of these resources, which are used in computer and information infrastructures, is very important. It is very effective in the management of the cost of these resources that the resource needs that may increase due to various reasons are rapidly expanded and narrowed when not needed. Flexibility in infrastructure costs is important, but it becomes troublesome as transactions are repeated. Even if there are various automation tools to meet the need, it does not always give the desired result when the configurations are considered. In such cases, solutions to meet the needs of the configuration should be created once only.

In the gains achieved, it has been explained that the installation and maintenance process, which consists of almost hundreds of stages, has been reduced to a few steps with the created scripts and the use of configuration tools, resulting in significant time savings.

BÖLÜM 1. GİRİŞ

Teknolojinin gelişmesi sosyal yaşamın tüm alanlarını büyük ölçüde etkilemiştir. Eğitim bu gelişmelerden etkilenen alanlardan biridir. (Akpınar, 2003) Uzaktan öğrenme, öğrencilerin birbirlerinden ayrıldığı ve kaynakları zaman veya mekân bağlamında öğrendiği, birbirleriyle ve öğrenme kaynakları ile etkileşimlerinin uzaktan iletişim sistemine dayandığı bir öğrenme sürecidir. (Özkul & Aydın, 2012) Meydana gelen hızlı gelişmelerin interaktif bilgisayarları ve internetin yaygınlaşması sonucunda eğitimin sıradan yöntemlerin yerine yeni yaklaşımları olduğunu ortaya çıkmıştır. E-öğrenmenin gün geçtikçe eğitimde rolünün arttığını gösteren araştırma, eğitim yönetim sistemi kullanımının belli kriterlere bağlı olması gerektiğini ispatlanmıştır. (Yıldırım, Göktaş, Temur, & Kocaman, 2004)

Günümüzde oldukça popüler hale gelen uzaktan eğitim konusunda destek veren birçok yazılım bulunmaktadır. Bu yazılımlardan bazıları hali hazırda konferans amaçlı kullanılırken bazılarının ise doğrudan uzaktan eğitim amaçlı kullanıldığı görülmektedir. Kurumlar bu faaliyetleri bir servis ile anlaşarak sürdürülebilirken bir diğer alternatif ise uzaktan eğitim yazılımını ve altyapısının kurumun kendi bünyesinde bulundurmaktır. Uzaktan eğitim faaliyetlerinin kendi bünyesinde bulundurulduğu modelde ilgili yazılımda kullanılacak altyapıların oluşturulması, ilgili yazılımına ait kurulum adımlarının tamamlanması, gerekli özelleştirmelerin yapılması uzun ve karmaşık bir süreçtir. Bu çalışmada seçilen bir açık kaynak kodlu uzaktan eğitim yazılımının baştan sonra tüm kurulum işlemlerinin otomatikleştirilmesi gerçekleştirilmiştir. Elde edilen zamanın yanı sıra bu yaklaşım taşınabilir ve felaket senaryolarında kolaylıkla yeniden oluşturulabilir.

Açık kaynak kodlu yazılımların kamuda kullanımı hem yazılımın güvenilirliği açısından hem de gider maliyetleri açısından avantajlıdır. Açık kaynak kodlu yazılım

belli grupların iş birliğiyle oluşturduğu, kaynak kodlarının herkes tarafından incelenebildiği ve koşullara göre gerekirse değiştirebildiği ve çoğunlukla ücretsiz kullanılabilen araçların bütünüdür. Özellikle kamu kurumlarında tercih edilen açık kaynak kodlu yazılımlar hem mahrem bilginin güvenli şekilde işlenmesi hem de yazılım giderlerinin düşürülmesi açısından çoğunlukla kullanılmaktadır.

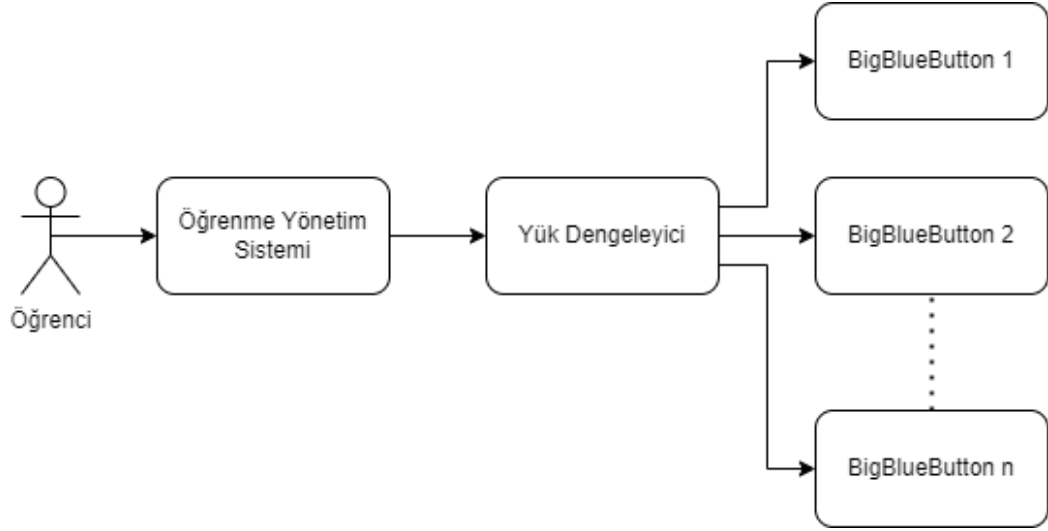
Bir uzaktan eğitim yazılımının altyapısı oluşturulurken önemli diğer konu ise bu yapının gözlemlenebilir olmasıdır. Yapının parçalarını oluşturan her bileşenden toplanan veriler hem planlamanın gelecekte optimize edilmesinde hem de elde edilen veriler görselleştirilerek kolay yorumlanması önemlidir. Her bir sunucuda oluşacak olağanüstü durumların alarmlar oluşturularak anında bildirilmesi, sistemin sürekliliğini sağlamak için gereklidir. Eğitim amacıyla kurulmuş bir çevrimiçi toplantı yazılımında ihtiyaca yönelik birçok ayrıntı düşünülmüş olmalıdır. Amaca hizmet eden toplantı yazılımının yanında öğrenim yönetim sistemi bu yapıyı tamamlayan bir yazılımdır.

Günümüzde işletim sistemlerinde özellikle Windows tarafında komut satırı çok kullanılsa da açık kaynaklı Linux çözümlerinde kurulum, bakım, yönetim gibi konularda terminal sıkça kullanılmaktadır. Düzenli aralıklarla kullanılması gereken komutların bir betik haline getirilmesi bu işlemleri bir hayli kısaltmaktadır. Tekrar eden aşamaların betik haline getirilmesi hem sistem yöneticisine vakit kazandırmaktadır hem felaket senaryolarında yapının tekrar ayağa kaldırılmasında kolaylık sağlamaktadır hem de kurulum aşamalarının gözden geçirilmesinde doküman desteği sağlamaktadır. Canlı kullanılmakta olan bir sistemin barındırdığı riskleri en aza indirmeyi amaçlayan bu betikler, var olan sistemin olası felaket senaryolarında en hızlı bir şekilde tekrardan işlevini yerine getirebilecek konuma alınmasında büyük kolaylık ve faydası bulunmaktadır. Bu çalışmada bir uzaktan konferans yazılımına ait tüm bileşenler ve ona ait tekrarlayan işlevlerin betikler, yardımcı configürasyon araçları ve kod olarak altyapı araçları kullanılarak yüzlerce sayılabilecek aşamayı tek haneye indirmek amaçlanmaktadır.

Bu alıřma tm ařamalarıyla hazırlanmıř aık kaynak kodlu bir evrimii toplantı ortamının yanında bu ortamın srekliĐinin saĐlanması, kurulum ařamalarının hazırlanması ve otomatikleřtirilmesi, sık karřılařılan problemlere etkin zm getirilmesi, uygun izleme yazılımlarıyla denetlenmesi ve bu yapıdan retilen verilerin akıllı yntemler ile iřlenmesini iermektedir.

BÖLÜM 2. İDEAL UZAKTAN EĞİTİM SİSTEMİ

Uzaktan eğitim altyapı bileşenlerinde, kurumun kendi geliştirdiği ticari çözümler ve yerel çözümlerin yanı sıra, dünya genelinde geniş kullanıcı potansiyeline sahip topluluklar tarafından geliştirilen ve desteklenen açık kaynaklı sistemler kullanılmaktadır. (Süral, Açık ve uzaktan öğrenmede teknolojik altyapının oluşturulması, 2015) Günümüzde birçok ülke açık kaynaklı yazılım kullanmaktadır. Ulusal kurumlar tarafından uygulanan proje ve politikalarla farkındalık oluşturulmaya çalışılmaktadır. Çevrimiçi öğrenme ortamı yazılımlarının örnekleri kıyaslanarak incelenen; yararları ve sınırlılıkları dile getirilirken lisanslama, paylaşım, video oynatımı ve dil desteği başlıkları altında ayrı ayrı analiz edilmiştir. Bütün yazılımların temel özelliklere sahip olduğu görülürken birbirlerine göre sınırları ve üstünlükleri de ayrıca keşfedilmiştir. (İzmirli & Akyüz, 2017) Birçok çalışmada açık kaynak kodlu yazılımlarda özel sektöre yönelik devlet teşvik politikalarının olduğu görülmektedir. (Akyıldız, Kamu Yönetiminde Açık Kaynak Kodlu Yazılımlar, 2012) Bu çalışmada uzaktan öğrenme platformu yazılımı olarak BigBlueButton ve ona destek olacak yan araçlar tercih edilmiştir. Uzaktan ve açık öğrenme hizmetlerinin teknolojik altyapı bileşenleri olarak öğrenme yönetim sistemleri ve bunun beraberindeki araçları tek tek açıklanmıştır. Teknolojik altyapının oluşturulmasının yanında ticari çözümlerle birlikte açık kaynak yazılımların yükseköğretimde kullanılabilirliğini gösteren çalışma, bu hizmeti sunan kurumlara ve onların yöneticilerine rehber olma amacı taşımaktadır. (Süral, Açık ve uzaktan öğrenmede teknolojik altyapının oluşturulması, 2015) Bu çalışmada çevrimiçi toplantı uygulaması olarak BigBlueButton ve ona yardımcı olacak araçlar tercih edilmiştir. BigBlueButton tek başına çevrimiçi öğrenim için yeterli olmamaktadır. İhtiyaç duyduğu araçlar ile birlikte gerekli yapı Şekil 2.1.'de gösterilmiştir.



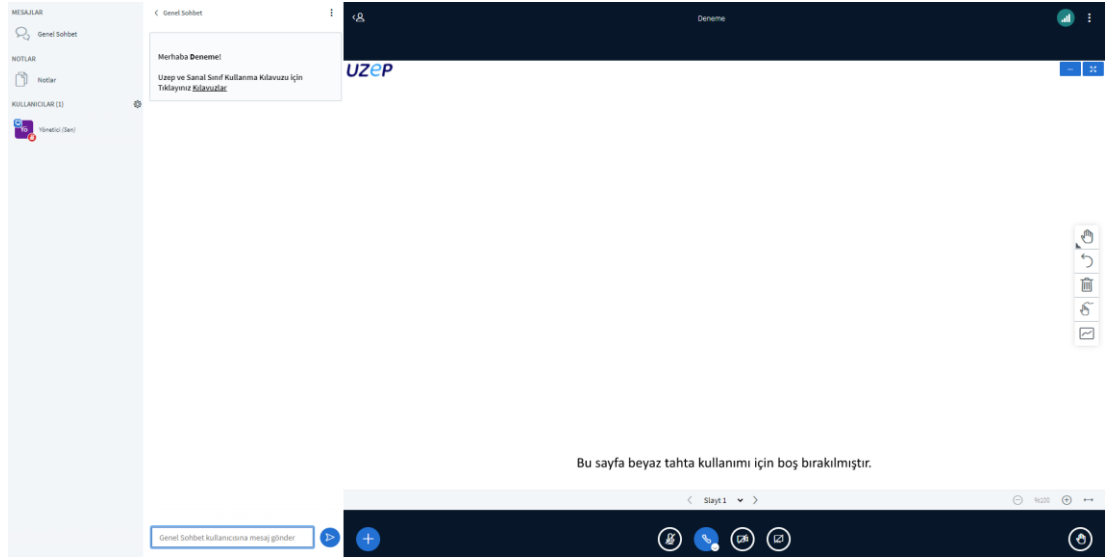
Şekil 2.1. İdeal BigBlueButton Yapısı

2.1. BigBlueButton

BigBlueButton, özellikle uzaktan eğitim için geliştirilmiş bir video konferans uygulamasıdır ve standart çevrimiçi toplantılarda da kullanılabilir. Uygulama, birden fazla kullanıcının aynı anda bir web kamerası kullanarak oturum açmasına ve aynı anda VoIP kullanarak birbirleriyle iletişim kurmasına olanak tanır. (Süral, 2011) BigBlueButton, uzaktan eğitim amaçlı açık kaynak kodlu bir web konferans yazılımıdır. Öğrencilere çevrimiçi öğrenim alanı sunmak için gerekli özellikleri bünyesinde bulundurur. Yazılım ayrıca belirli amaçlar için şu araçları kullanır;

- Freeswitch: Ses aktarımı sağlamak için kullanılır.
- Kurento: Görüntü ve ekran paylaşımı sağlamak için kullanılır.
- LibreOffice: Dokümanların yüklenmesi için kullanılır.
- NGINX: Ters vekil sunucusu olarak çalışır.
- Redis ve MongoDB: Çeşitli önbellekleme işlemleri için kullanılır.

BigBlueButton oturumunda öğretmenler ve öğrenciler interaktif ortamda toplantılarını gerçekleştirebilir. İçeriğinde eğitimi desteklemek için bir çizim tahtası, yazı yazma alanı, anket fonksiyonu, ekran paylaşma imkânı ve doküman yükleme fonksiyonu bulunmaktadır.



Şekil 2.2. BigBlueButton kullanıcı arayüzü

BigBlueButton yazılımını alternatiflerinden ayıran en önemli nokta tamamen eğitim odaklı geliştirilmiş olmasıdır. Kullanıcılarına çevrimiçi iletişim sunmasının yanında eğitim odaklı kullanılmak için çeşitli özelliklere sahiptir. Barındırdığı beyaz tahta sayesinde üzerine notlar alınabilmekte, anket fonksiyonu sayesinde kullanıcıların tercihleri takip edilebilmekte, yüklenen dokümanlar kullanıcılar eşliğinde incelenebilmektedir. İlgili yazılımın arayüzü Şekil 2.2.'deki gibidir. Açık kaynak kodlu olması sayesinde kolaylıkla kurum özelinde değişiklikler yapılabilmektedir.

2.2. Öğrenme Yönetim Sistemi

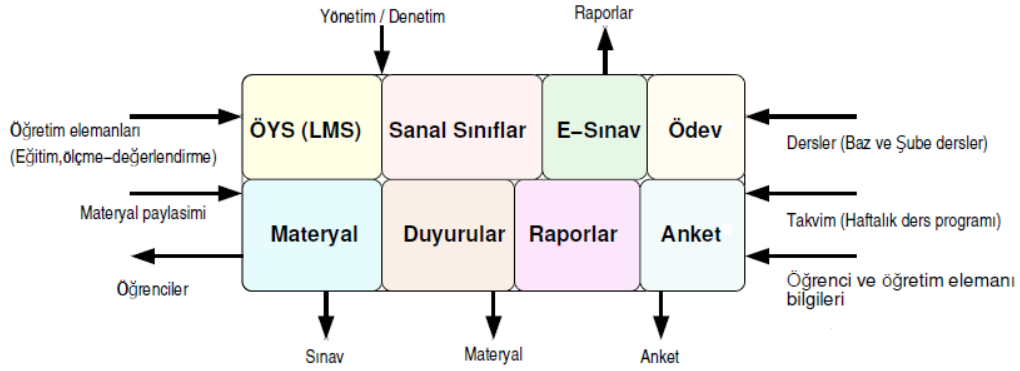
Sanal öğrenme ortamı veya öğrenme platformu olarak da bilinen uzaktan eğitimde öğretim yönetim sistemi önemli bir yer tutmaktadır. Öğrenme yönetim sistemi, öğrenme sürecini planlayabilen, değerlendirebilen ve uygulayabilen bir yazılım veya web tabanlı teknoloji olarak tanımlanabilir. Genel olarak, öğretmenlere eğitim içeriklerini elektronik formatta hazırlama ve yönetme, ayrıca öğrencilerin materyalleri kullanma performanslarını değerlendirme ve katılımlarını izleme fırsatı veren öğrenme yönetim sistemleri kullanılmaktadır. Açık kaynak öğrenme yönetim sistemleri, GNU lisansı altında dağıtılan yazılımları içerir. (Aydın & Boroğul, 2008)

Bunlardan bir örneği olan Moodle, açık kaynak kodlu oluşturulmuş bir öğrenme yönetim sistemidir. Bu sistemin kısaltması, esnek bir nesne yönelimli dinamik öğrenme ortamı olan Modular Object Oriented Dynamic Learning Environment'dir. Herkesin Açık kaynak internet tabanlı bir sistem geliştirebilmesi, zaman içinde kullanıcı tabanını de artırmıştır. (Uygun, Yıldız, & Hertsch, 2010) Tarama modelini kullanan bir çalışmada Türkiye'deki 208 üniversiteyi kapsamaktadır. Araştırma sonucunda elde edilen bulgulara göre en çok kullanılan öğretim yönetim sistemleri Moodle ve ALMS olduğu görülmüştür. Bunun yanında en çok kullanılan çevrimiçi toplantı yazılımının ise BigBlueButton ile Perculus olduğu tespit edilmiştir. (Durak, Çankaya, & İzmirli, 2020)

Çalışmada tercih edilen bir diğer öğrenme yönetim sistemi olan UZEP, Tübitak-1001 projesi desteğiyle geliştirilmiş ve Türkiye genelindeki çoğu üniversite tarafından kullanılmaya başlanmış bir uzaktan eğitim platformudur. Sistemin donanım altyapısı ULAKBİM tarafından sağlanmakta olup, yazılım bileşenleri tamamen Sakarya Üniversitesi bünyesinde geliştirilmiştir. UZEP platformunun barındırdığı fonksiyonların şeması Şekil 2.3.'teki gibidir. Geliştirilen sistem üniversitelerin bilgi kaynaklarına ihtiyaç duymadığından kısa sürede herhangi bir üniversitede kullanıma açılabilir. (UZEP uzaktan eğitim platformu, 2021)

- Öğrenme Yönetim Sistemi (ÖYS) Modülü: Haftalık sanal ders programlarının oluşturulmasını, uzaktan yürütülen eğitim faaliyetlerinin organize edilmesini ve yürütülmesini sağlayan modül.
- Sanal Sınıf Modülü: Sanal sınıfların interaktif ortamını sağlayan ve açık kaynak yazılımlardan yararlanılarak oluşturulmuş bileşen.
- Duyuru Modülü: Öğrencilere çeşitli seviyelerde duyuru yapılabilmesini sağlayan bileşen.
- Çevrimiçi Sınav Modülü: Çoktan seçmeli, ucu açık, doğru yanlış tarzda soruların hazırlanabildiği ve sınavların yürütülebildiği bileşen.
- Doküman Paylaşım Modülü: Ders kapsamında öğrencilerle her türlü dijital belge paylaşımının sağlandığı bileşen.

- Raporlama Modülü: Öğrenci devamı, akademisyenlerin sanal derse oranı, yürütülme sayıları gibi birçok raporun sistem üzerinden oluşturulduğu bileşen.
- Ödev Modülü: Araştırma ve kısıtlı süreli ödevlerin verildiği ve sistem üzerinden toplandığı bileşen.
- Anket Modülü: Derslere ait memnuniyet anketlerinin yapılabildiği bileşen.



Şekil 2.3. UZEP fonksiyonlarının şeması (UZEP uzaktan eğitim platformu, 2021)

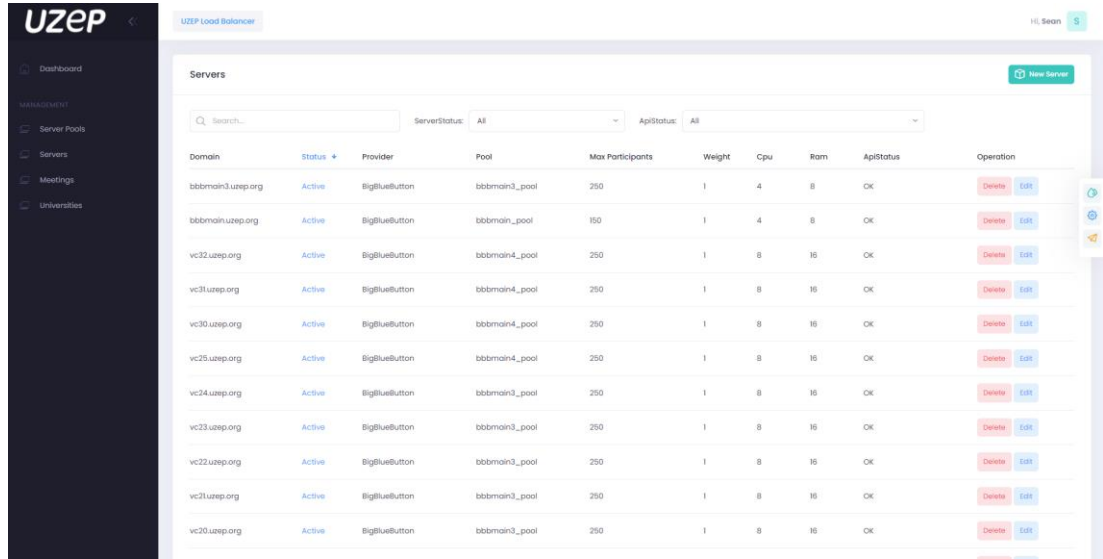
2.3. Yük Dengeleyici

Her yazılım ve donanımın belirli bir kapasitesi vardır. Bu kapasitenin fazlasına ihtiyaç duyulduğunda ya ilgili yazılım veya donanımın kapasitesi yükseltilir ya da aynı işlevi gören eşleri yapıya eklenir. Aynı işlevi gören donanım veya yazılımların bu kapasiteyi adil bir şekilde kullanması için bir yük dengeleyiciye ihtiyacı vardır. Yük dengeleyici kendi üzerine gelen istekleri bağlı olduğu donanım veya yazılıma adil olarak dağıtır. Yük dengeleyici yazılım kullanmanın birçok amacı vardır. Bu çalışmada sadece BigBlueButton aracına yardımcı yük dengeleyici yazılımlardan bahsedilecektir.

Scalelite, BigBlueButton sunucu havuzunu yönetmek için tasarlanmış bir yük dengeleyicidir. Birden fazla BigBlueButton sunucunun tek bir sunucu gibi gözükmesini sağlayarak arka planda yük dengeleme işlemi yapar. Her BigBlueButton sunucusuna API isteği yapmak yerine istekler Scalelite üzerinden yapılır ve Scalelite, BigBlueButton sunucularının uygunluk durumuna göre gerekli yönlendirmeyi yapar. Ortalama kapasitede bir BigBlueButton sunucusu aynı anda yaklaşık 150 kullanıcıya hizmet verebilirken kaynakların dikey olarak artırılması bu kapasiteyi network trafiği

kısıtından dolayı yukarı taşıyamaz bu nedenle yatay genişleme ve bu genişleme için bir yük dengeleme sistemi gereklidir.

Scalelite yük dengeleyicisinin eksikleri göz önünde bulundurularak Sakarya Üniversitesi bünyesinde oluşturulan UZEP yük dengeleyici, alternatifi olan bir öncekine göre en büyük avantajı bir görsel arayüz sunmasıdır. Yönetilen BigBlueButton sunucu sayısı ve onun üzerinde yer alan toplantıların sayısı arttıkça ortaya bir yönetim sorunu çıkmaktadır. Bu denli karmaşık bir yapıyı komut satırından yönetmek zorlaştığından dolayı UZEP yük dengeleyici yazılımı bir WEB arayüzüyle bahsi geçen zorluğa çözüm olmuştur. Yazılımın WEB arayüzü Şekil 2.4.'te gösterilmiştir. UZEP yük dengeleyici yazılımının öne çıkan diğer fonksiyonları ise mevcut toplantıların listelenmesi ve gerektiğinde yönetici rolüyle katılabilme, video dönüştürülme işlemi süren kayıtların son durumlarının izlenmesi, farklı topluluklara ait havuzların mantıksal olarak ayrılabilmesi sayılabilmektedir. Bu yazılım havuzunda bulunan sunucuları belli aralıklarla kontrol ederek problem olan sunucuları giden yönlendirmeleri otomatik olarak kapatabilmektedir. Eğer bakım veya güncelleme yapılacak bir sunucu var ise bu yönlendirme işlevi manuel olarak da kapatılabilir.



Domain	Status	Provider	Pool	Max Participants	Weight	Cpu	Ram	ApiStatus	Operation
bbbmmain3.uzep.org	Active	BigBlueButton	bbbmmain3_pool	250	1	4	8	OK	Denkle Ekle
bbbmmain.uzep.org	Active	BigBlueButton	bbbmmain_pool	150	1	4	8	OK	Denkle Ekle
vc32.uzep.org	Active	BigBlueButton	bbbmmain4_pool	250	1	8	16	OK	Denkle Ekle
vc31.uzep.org	Active	BigBlueButton	bbbmmain4_pool	250	1	8	16	OK	Denkle Ekle
vc30.uzep.org	Active	BigBlueButton	bbbmmain4_pool	250	1	8	16	OK	Denkle Ekle
vc25.uzep.org	Active	BigBlueButton	bbbmmain4_pool	250	1	8	16	OK	Denkle Ekle
vc24.uzep.org	Active	BigBlueButton	bbbmmain3_pool	250	1	8	16	OK	Denkle Ekle
vc23.uzep.org	Active	BigBlueButton	bbbmmain3_pool	250	1	8	16	OK	Denkle Ekle
vc22.uzep.org	Active	BigBlueButton	bbbmmain3_pool	250	1	8	16	OK	Denkle Ekle
vc21.uzep.org	Active	BigBlueButton	bbbmmain3_pool	250	1	8	16	OK	Denkle Ekle
vc20.uzep.org	Active	BigBlueButton	bbbmmain3_pool	250	1	8	16	OK	Denkle Ekle

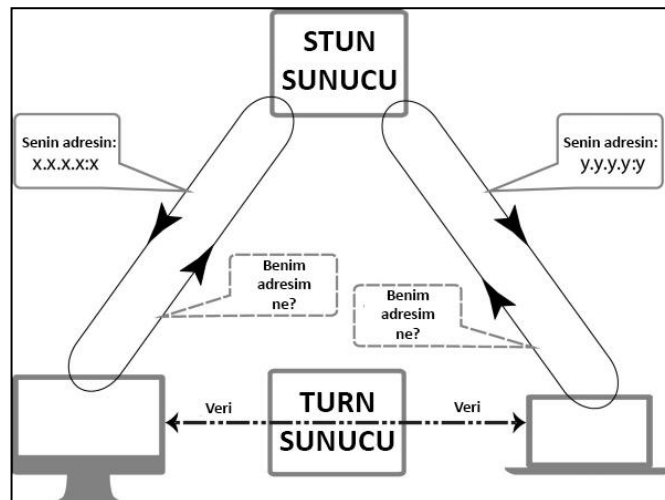
Şekil 2.4. UZEP yük dengeleyici WEB arayüzü

2.4. Yardımcı Araçlar

BigBlueButton yazılımı tüm fonksiyonlarını yerine getirmek için birtakım araçlara ihtiyaç duyar. Kendisi birçok açık kaynaklı yazılımın bir araya getirilmesi ile oluşmuş olsa dahi dışardan da bazı araçlarla desteklenmelidir. Bu araçlardan kimisi ağ bağlantısının oluşmasına yardımcı olurken kimileri de kayıtların saklanmasına ortam hazırlamaktadır. Yardımcı araçlar, adından da anlaşılacağı gibi BigBlueButton yazılımının çalışmasını kolaylaştırmasına karşın kullanımı zorunlu değildir.

2.4.1. TURN ve STUN sunucu

Ses ve görüntü kullanan medya sunucularının güvenlik duvarı veya NAT arkasından erişimini sağlamak için gerekli olan araçtır. Paketleri eşler arasında aktarmak için bir eş sunucu görevi yapar. STUN sunucu, NAT arkasında bulunan sunuculara ait UDP paketlerinin iç IP verisini dış IP verisi ile değiştirerek sağlıklı bağlantı kurulmasına yardımcı olur. TURN sunucu, STUN sunucunun dahi yeterli olmadığı durumlarda kullanılır. Eşlerden ikisi de TURN sunucuya bağlanması yoluyla bir ara sunucu görevi yapar. TURN sunucusunun çalışma mantığı Şekil 2.5.'teki ifade edilmiştir. TURN ve STUN sunucuların yapılandırılması sağlıklı BigBlueButton kullanımı için gereklidir.



Şekil 2.5. TURN ve STUN sunucuların çalışma mantığı.

2.4.2. NFS dosya paylaşımı

Ağ Dosya Sistemi, dosyaları bir ağ üzerinde depolamak için bir mekanizmadır. Kullanıcıların uzak bilgisayarlarda bulunan dosya ve dizinlere erişmesine ve bu dosya ve dizinlere yerelmiş gibi davranmasına izin veren dağıtılmış bir dosya sistemidir. (IBM, 2020) Birden çok sayıda bulunan BigBlueButton sunucularında elde edilen toplantı kayıtlarının bir noktada depolanması gerekmektedir. Ağ dosya sistemi bu BigBlueButton sunucularında çalışarak ilgili video kayıtlarının düzenli aralıklarla aktarılmasına yardımcı olur.

2.5. Sunucuların İzlenmesi

Eşzamanlı çalışan BigBlueButton sunucularından ve üzerinde gerçekleşen çok sayıda toplantıdan değerli veriler elde edilebilmektedir. Bu verilerin sınıflandırılarak değerlendirilmesi yapının oluşturulmasındaki karar sistemlerine yardımcı olabilirken toplantı verilerinin incelenmesi bu toplantıların verimliliği hakkında bize bilgiler verebilmektedir. Çalışma içerisinde sunucu kaynakları ve toplantı verileri olarak ikiye ayrılan verilerin değerlendirilmesi başlığı, sistem yöneticileri ve eğitmenlere ayrı ayrı fikir verebilmektedir. Sunucu kaynaklarından elde edilen veriler sunucuların işlemci kullanımı, bellek kullanımı, disk doluluğu, üzerlerinde yürütülen toplantı sayısı gibi verileri bize çeşitli ajanlar vasıtasıyla sunabilmektedir. Toplantı aktivitelerinden elde edilen veriler BigBlueButton yazılımının bir özelliği olarak doğrudan görsel olarak değerlendirilebilirken, bu veriler daha derin çıkarımlar yapmak için manuel olarak da değerlendirilebilir. Bu çalışmada hem sunucu kaynaklarından elde edilen verilerin anlamlandırılarak görselleştirilmesi hem de toplantı verilerinin toplanarak çeşitli faktörlere göre değerlendirilmesi araştırılmıştır.

Süreklilik sağlayan bir yazılım ve donanım topluluğunda bu durumu korumak için normal dışı durumların düzenli aralıklarla kontrolü şarttır. Aralıksız çalışan sistemlerde belli bakım fonksiyonları düzenli olarak yürütülse de kontrol dışı gerçekleşen durumlardan ötürü sistemin devre dışı kalmasının belli maliyetleri olabilir.

Sürekli hizmet vermesi zorunlu olan sistemlerin takip edilmesi ve limitlerinin belirlenmesi zorunluluktur.

Donanım ve yazılımın bütünü oluşturulan sistem küçük sayılarda kontrol edilebilir ancak yapı büyüdükçe onlara ait herbir parametreyi takip etmek güçleşir. Bu durumda yardımcı ajan yazılımlar sistem durma noktasına gelmeden veya durduktan çok kısa süre içerisinde müdahale etme şansı vermektedir. Bu yazılımlar tek başına kontrolü sağlayıp kullanıcıya sunabildiği gibi çeşitli alarm fonksiyonları vasıtasıyla ortam yöneticisini uyarabilir ve felaket senaryolarının önüne geçilmesinde yardımcı olabilmektedir.

2.5.1. Zaman serisi veritabanı

Zaman serisi, belli bir zaman aralığında kaydedilen verilerin bir araya toplanmasıdır. Zaman serisi veritabanları düzenli olarak topladığı verileri tarih ve saatleriyle kaydeder. Gözlemlenmek istenen yapının belirli zaman aralıklarında durumunu incelemek için kullanılır.

Sistem yöneticilerinin izleme amaçlı kullandığı açık kaynaklı Prometheus isimli yazılım bir zaman serisi veritabanıdır. Kaynaklardan topladığı verileri içerisine düzenli olarak kaydeden bu veritabanı, kendine has PromQL dili ile bu verileri istenilen nitelikte getirebilir. Çok sayıda görselleştirme aracı ile entegre çalışabilen bu veritabanı, aynı zamanda verilerin görselleştirilmesine aracılık etmektedir.

2.5.2. Ajan yazılımlar

Sunuculardan verileri dışarı aktarmak için kimi ajan yazılımlara ihtiyaç duyulmaktadır. Bu yazılımlar sunucu üzerindeki işlemci kullanımı, disk kullanımı, bellek doluluk oranı gibi temel verileri çeşitli protokoller aracılığıyla dışarı aktarmaktadır. Ajan yazılımlar amaçlarına göre çeşitlere ayrılmaktadır. Bu çalışmada sistem donanımlarını izlemek ve BBB yazılımının durumunu izlemek için iki farklı ajan yazılım kullanılmıştır. Verileri dışarı aktarmak için HTTP protokolünü kullanan bu

ajanlar kendilerine istek yapıldığı zamanda yükümlü olduğu verileri geri döndürmektedir. Prometheus veritabanına ajan yazılımların adresleri tanımlandığında düzenli aralıklarla veri aktarımı gerçekleştirilerek toplanan verilerle efektif izleme sağlanabilmektedir.

2.5.3. Verilerin görselleştirilmesi

Kaynakları izlemek için gerekli veriler, onları değerlendirecek bir arayüz olmadan bir işe yaramamaktadır. Ham halde çekilen veriler birtakım işlemler ile birlikte anlamlı hale getirilebilir ve bu işlenen veriler çeşitli araçlar yardımıyla görselleştirilebilmektedir. Görselleştirilen veriler aracılığıyla sunucu kaynakları ve yazılımın durumunu kolaylıkla yorumlanabilmektedir.

Sunucu kaynakları ve yazılımların durumunu görsel olarak izlemek için çok sayıda araç bulunmaktadır. Bu araçlardan kimileri hazır şablonları kullanırken kimileri ise elde edilen verilerin niteliğine göre büyük esneklik sunmaktadır.

Bu çalışmada verileri görselleştirmek için açık kaynak kodlu Grafana yazılımı tercih edilmiştir. Bir görselleştirme aracı olarak Grafana, yığınları izlemede çok popüler bir bileşendir ve genellikle zaman serisi veritabanları ve izleme platformlarıyla birlikte kullanılır. (Sierra-Fernández, ve diğerleri, 2021) Grafana aracının sunduğu görsel veriler Şekil 2.6.'da gösterilmiştir. Araç, Prometheus veritabanından aldığı verileri bu veritabanına has PromQL diliyle istenilen şekilde çekip çeşitli grafik ve görselleştirme imkânı vermektedir. Elde edilen çok sayıda veriden görselleştirilenler şunlardır;

- Toplam katılımcı sayısı.
- Görüntülü katılımcı sayısı.
- Sesli katılımcı sayısı.
- Toplam toplantı sayısı.
- İşlemci kullanımı.
- Bellek kullanımı.
- Disk kullanımı.



Şekil 2.6. Grafana aracı ile verilerin görselleştirilmesi.

2.5.4. Toplantı aktivite verileri

Dünyanın dört bir yanındaki yükseköğretim kurumları, daha etkili ve verimli bir öğrenme süreci elde etmek için öğrenme analizi teknolojisini her geçen gün daha fazla kullanmaktadır. Küresel rekabet ortamındaki hızlı artış ve karmaşıklık ile yükseköğretim kurumları giderek yeni dinamiklerle karşı karşıya kalmaktadır. Eğitimde harmanlanmış ve çevrimiçi öğrenmeye mevcut geçiş ve öğretim yönetim sistemlerinin tanıtılmasıyla birlikte, büyük verinin eğitimde bir yeri olması şartı değil. Ayrıca önümüzdeki birkaç yıl içinde yükseköğretim kurumlarının yoğun büyük veri uygulamaları yapacağı öngörülmektedir. (Tutsun, 2020)

BigBlueButton yazılımı, toplantı analitiklerini geçici bir Redis önbelleğinde tutar. Toplantı kayıtları işlendikten sonra silinen bu kayıtlar eğitim analitiklerini araştırmak için oldukça değerlidir. Bu veriler, toplantı kayıtları yeniden izlenmek üzere oluşturulurken yerleşim düzenini koruyabilmek için kullanılmaktadır. Bu verilerin içerisinde saklanan değerler şu şekildedir;

- Katılımcıların toplantıya giriş-çıkış verileri.
- Toplantı başlangıç ve bitiş zamanı.
- Katılımcıların yazdığı mesajlar.
- Katılımcıların mikrofon ve kamera açma-kapama zamanları.
- Açılan anketler ve katılımcılar tarafından verilen oylar.

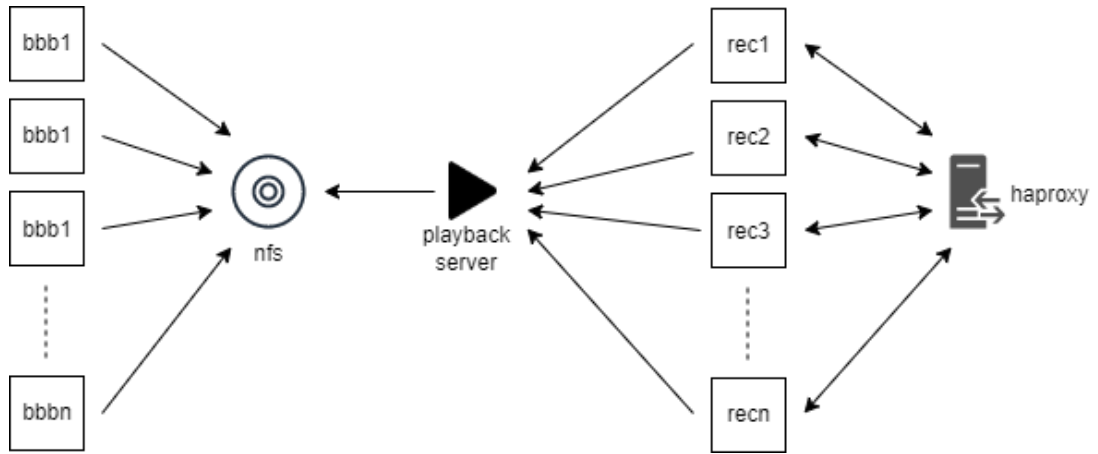
Normal davranışta BigBlueButton topladığı geçici verileri belirlenen süre sonunda silmekte ve herhangi bir analitikte kullanmamaktadır. Bu çalışma kapsamında, geleceğe yönelik bu verilerin kullanılabilmesi için oluşturulan bir betik yardımıyla içeri yedeklenmesi sağlanmaktadır. Açık kaynak kodlu bu yazılımın henüz kullanıma açılmamış test sürümünde bu analitik verileri JSON dosya formatında saklanıp hazırlanmış bir arayüzle toplantı sahibine sunmaktadır. JSON, metin temelli, neredeyse bütün programlama dilleri tarafından kullanılabilir veri aktarımı için kullanılan bir yazım biçimidir. (Ayaz, 2019) Birçok programlama dili bu yazım biçimini desteklemesinden ötürü farklı ortamlarda bu verileri değerlendirmek amacıyla kalıcı olarak saklanması sağlanmıştır.

2.5.5. Toplantı kayıtlarının işlenmesi

Çevrimiçi toplantılarda gerçekleştirilen bir diğer genel alışkanlık ise toplantı kayıtlarının alınmasıdır. Klasik öğrenme yöntemlerinin aksine çevrimiçi öğrenim yöntemlerinde öğretmenler ve öğrenciler tarafından daha sonra erişebilmek üzere toplantı kayıtları talep edilmektedir. BigBlueButton yazılımı bünyesinde gerçekleştirilen toplantıların kaydedilmesine olanak sağlamaktadır. Kaydedilen toplantılar belirli işlemlerden geçirilerek son haliyle kullanıcılara ulaştırılmaktadır. BigBlueButton yazılımı varsayılan olarak kayıtları yalnızca kendi oynatıcısı aracılığıyla yürütmektedir. Taşınabilirliği kısıtlayan bu problem bazı yeni yöntemler geliştirmeye itmek zorunda bırakmıştır.

Toplantıların yalnızca özel bir oynatıcı vasıtasıyla yürütülebilmesi, kayıtların saklanması ve farklı ortamlara aktarılmasına engel olmaktadır. Bu problemin önüne geçebilmek için kayıtların belirli medya formatlarıyla oynatabilmek amacıyla çeşitli işlemlerden geçirilmesi gerekmektedir. Kullanılabilecek ilk yöntem ders kayıtlarının ekran kaydı vasıtasıyla saklanmasıdır. İkinci yöntem ise kayıt medyalarının çeşitli işlemlerden geçirilerek içerisine ses ve yazıların gömülmesidir. İki yöntemin de birbirine göre avantajları ve dezavantajları vardır. Yapılan çalışmada her iki yöntem de denenmiş olup birbirine göre üstünlükleri tartışılacaktır.

Toplantıların izlenerek ekran kaydının alınması basit bir yöntem olsa da kullanılabilirlik ve zaman bakımından dezavantajları bulunmaktadır. Toplantı kaydının başlatılması için butona basılması ve ekran kaydının alınması belirli aşamaları barındırmaktadır. Bunun yanında bir toplantı kaydının medya dosyasına bu yöntem vasıtasıyla dönüştürülmesi minimum bir toplantı uzunluğu kadar sürmektedir. Geliştirilen yapıda çalışan betik kendisine parametre olarak iletilen toplantı kaydını otomatik olarak açarak ekran kaydını başlatmaktadır. Aynı anda birden fazla toplantıyı kaydedebilmek ve ortam bağımlılıklarından kurtulmak amacıyla için betik topluluğu bir konteyner haline getirilmiştir. Düzenli olarak gelen emirleri biriktirebilmek ve sıraya dizmek amacıyla kayıt sunucusunun önüne açık kaynak kodlu bir kuyruk yazılımı kullanılmıştır. Aynı anda birden fazla sunucuda bu hizmeti verebilmek için bir ters vekil sunucusu eklenerek emirlerin adil olarak kuyruğa dağıtılması sağlanmıştır.



Şekil 2.7. Kayıt medyalarının oluşturulması.

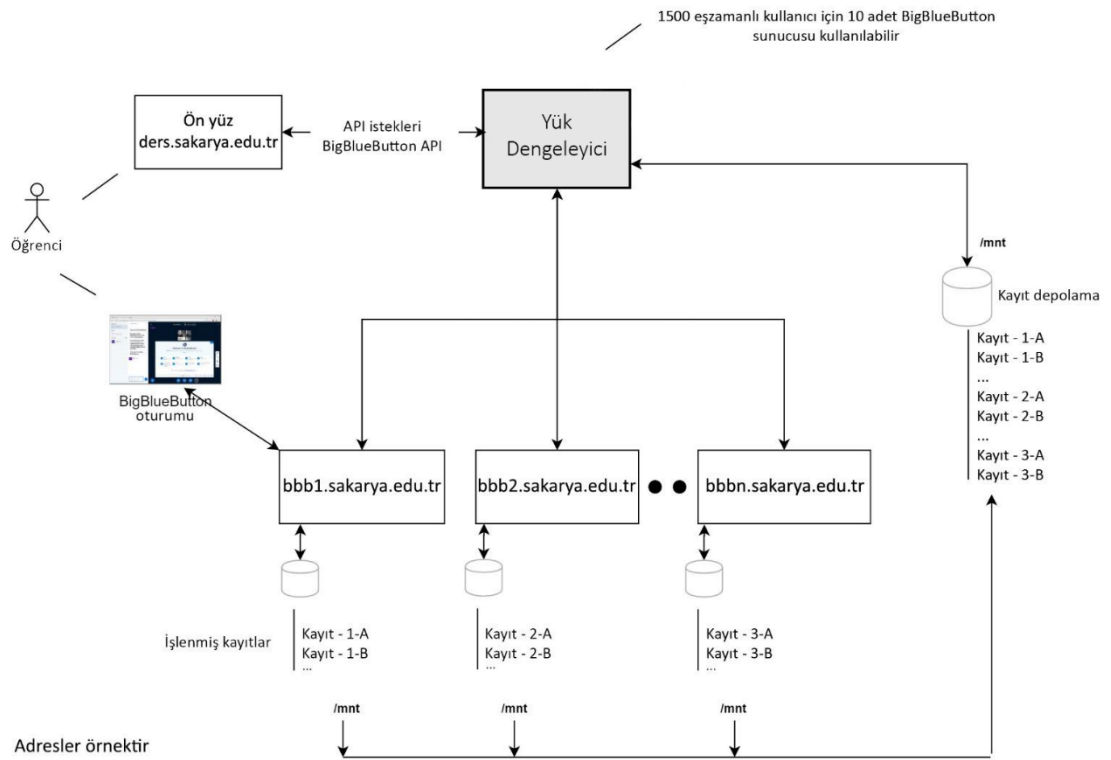
Geliştirilen yöntemle birlikte toplantı tamamlandıktan sonra ilk işlenmesi yürütülmekte ve işleme bittikten sonra BigBlueButton tarafından ters vekil sunucusunun medya dönüştürmesini tetiklemek için komut gönderilmektedir. Oluşturulan yapı Şekil 2.7.'de gösterilmektedir. Komut iletildikten sonra uygun sıraya göre kuyruğa alınan komut sırası geldiğinde sanal bir ortamda toplantıyı izleyerek kayıt altına almaktadır.

BÖLÜM 3. UZAKTAN EĞİTİM SİSTEMİNİN KURULUMU

BigBlueButton kurulumu, geliştirme ekibinin sunduğu bbb-install.sh betiği sayesinde kolaylıkla kurulmaktadır ancak yazılımın kendi dışında başka bağımlılıkları olmasından dolayı tek başına yeterli olmamaktadır. Bütünüyle çalışan sistemin tüm kurulum aşamaları şunlardır;

- BigBlueButton kurulumu.
- Yük dengeleyici kurulumu.
- NFS depolama kurulumu.
- TURN sunucu kurulumu.
- Gerekli özelleştirmelerin yapılması.
- İzleme araçlarının kurulumu.
- Öğrenme yönetim sisteminin kurulumu.

BigBlueButton kurulumu tek başına oldukça kolay bir kurulumla sahip olsa da sahip olduğu bağımlılıklar nedeniyle sağlıklı kullanım için gerekli tüm aşamaların tamamlanması bir hayli karışık olmaktadır. Tüm kurulum aşamalarının otomatikleştirilmesi operasyonların hızlıca gerçekleştirilmesi açısından oldukça önemlidir. Otomatikleştirilen aşamalar arasında gerekli altyapıların oluşturulması, BigBlueButton kurulumu ve gerekli bağımlılıkların oluşturulmasıdır. Tam fonksiyonlu bir BigBlueButton destekli uzaktan eğitim sisteminin mevcut yapısı Şekil 3.1.'de gösterilmiştir.



Şekil 3.1. Mevcut yapı. (Scalelite, 2022)

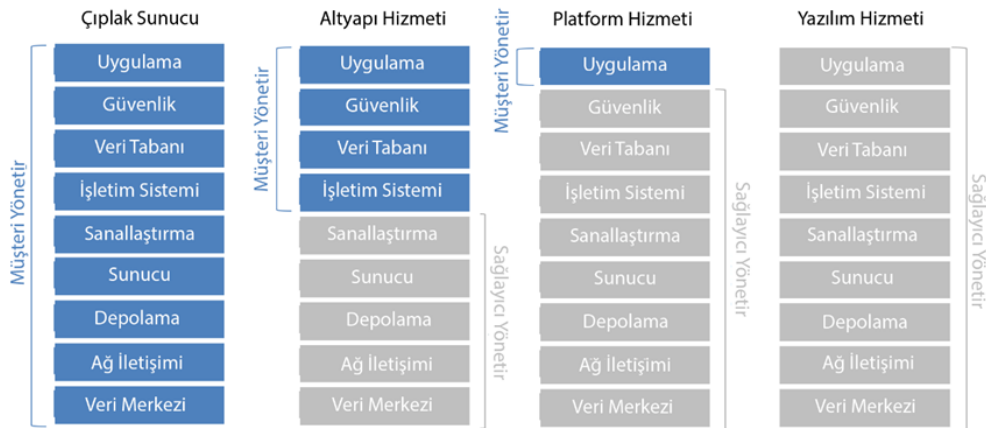
3.1. Altyapı Tercihii

Bir uzaktan eğitim sistemi oluşturulmasında en büyük giderlerden birisi de altyapı maliyetleridir. BigBlueButton yazılımı, planlanan aynı zamanlı öğrenci sayısına göre ciddi miktarda kaynak gereksinimi duyabilir. Bu kaynakların sağlanmasında kuruluş bünyesindeki donanımlar tercih edilebilirken artık bulut teknolojisi de öne çıkmıştır. Donanımların kurum içerisinde barındırılması, bakım ve sarf giderleri hesaba alındığında kimi durumlarda bulut üzerinden donanım kiralamak daha avantajlı olabilmektedir.

Önerilen özelliklerde (8 vCPU 16 GB bellek) bir BigBlueButton sunucusu yaklaşık toplamda 300 katılımcıyı eş zamanlı olarak misafir edebilmektedir. Yeterli gelmeyen durumlarda yanına eklenecek BBB sunucuları ihtiyaç duyulan yükü aralarında bir yük dengeleyici vasıtasıyla paylaşabilmektedirler. Yapılan çalışmada bulut üzerinde kurulum ve bu kurulumun otomatize edilmesi araştırılacaktır.

3.1.1. Bulut teknolojisi

Hızla gelişen bilgisayar teknolojisi ve artan kullanıcı sayısı birçok problemi beraberinde getirmiştir. Uygulamaların arkasında çalışan bilgisayar sistemleri taleplere cevap veremez duruma gelmiş, kaynak ölçeklemesi yetersiz bir hal almıştır. Bunun yanında donanımları güncel tutmak adına düzenli olarak yenilemek, bakımlarını yapmak, iklimlendirme giderleri, elektrik giderleri, güvenlik giderleri gibi maliyetler işletmelere yük olmaya başlamıştır. Bu problemler sektörde yeni arayışlara neden olmuştur. Bulut bilişim modeli çözümlerden biridir. Gerekli donanım üçüncü bir sağlayıcı tarafından sunulan, uzakta bulunan ve web üzerinden kontrol edilebilen, yönetim çabasının çok daha az olduğu bilişim modelidir. Bulut hizmet modelleri Şekil 3.2.'de görselleştirilmiştir. Sunulan donanımın yenileme, bakım, elektrik, yönetim, güvenlik gibi problemlerinden sağlayıcı sorumludur. Genellikle sistemin çalıştığı saat bazında ücretlendirilen yapı sunucu sistemini kendi bünyesinde bulundurmaktan daha karlı olmaktadır.



Şekil 3.2. Bulut hizmet modelleri.

3.1.1.1. Altyapı bulut hizmeti

Sağlayıcının kullanıcıya sadece donanım sunduğu modeldir. Altyapı masraflarından büyük oranda tasarruf edilir ancak geri kalan yazılım ve servislerin sorumluluğu kullanıcıdadır. Kullanıcı, sağlayıcının donanım, depolama, ağ altyapısı ve sanallaştırma imkanlarından faydalanır.

3.1.1.2. Platform bulut hizmeti

Sağlayıcının kullanıcıya donanımın yanında çalışacağı işletim sisteminin ve gerekli ara yazılımların da yani tam olarak platformun sunulduğu hizmet modelidir. Gerekli olan birden çok altyapının bir araya gelmesiyle oluşur. Altyapı modeline göre makineler hızlıca kullanmaya uygun hale getirilir ve zamandan tasarruf sağlanır.

3.1.1.3. Yazılım bulut hizmeti

Kullanıcıya donanım, işletim sistemi, sanallaştırma, ağ altyapısı, kullanılacak yazılım imkanlarının hepsinin sunulduğu hizmet modelidir. Yönetim tamamen sağlayıcı elindedir ve kullanıcıya çok az sorumluluk düşer. Hepsi bir arada çözümdür ancak ihtiyaca göre esnetilmesi daha güçtür. Sağlayıcının sunduğu özellikler ile yetinilir. Yönetim, kurulum, bakım, yazılım ihtiyacı gibi tüm sorumluluklar sağlayıcı tarafından üstlenilir.

3.2. Uygun Ortamın Hazırlanması

Çevrimiçi toplantı yazılımı BigBlueButton, çalışabilmek için birtakım bağımlılıklara ihtiyaç duyar. Bu bağımlılıklardan en önemlisi birçok açık kaynak kodlu yazılımlardan aşına olunduğu üzere temelde bir Linux işletim sistemidir. Linux işletim sistemlerinin çeşitli dağıtımları bulunmaktadır ancak geliştirici ekibin önerdiği dağıtım Ubuntu işletim sistemidir. Temelde tek bir sunucuda dahi çalışabilen BigBlueButton yazılımını etkin kullanılmak istendiğinde birçok ek bileşene ihtiyaç duymaktadır. Çok daha fazla sayıda kullanıcıya hizmet verilmesi gerektiğinde sayılarının artması gereken yazılım sunucusu, aynı zamanda bu sunucuların önüne bir yük dengeleyici eklenmesi ihtiyacını doğurmaktadır. Elde edilen kayıtların ortak bir noktada toplanması için sunucuların her birine bağlı bir ağ depolaması yine ihtiyaçlardan biridir. Ağ depolama sunucusu olarak önerilen tercih NFS (Network File System) protokolüdür.

3.3. Açık Kaynak Kodlu Yazılım

Geliştirilen yazılıma ait kaynak kodlarının herkes tarafından erişilebilmesi onun açık kaynak sıfatına sahip yapar. Gönüllüleri tarafından destek verilen projeler herkes tarafından incelenebilir ve böylelikle yazılımın güvenilirlik problemleri ortadan kalkmış olur. Birçok yazılım firması kaynak kodlarını mahrem olarak görse de günümüzde açık kaynağa yönelim artmıştır. Özgür yazılım kavramıyla karıştırılmaması gereken açık kaynak olgusu yazılımın ücretli dağıtılmasına engel değildir.

Açık kaynak kodlu yazılımlar günümüzde birçok ülkede kullanılmaktadır ve farkındalığı devlet kurumlarının üstlendiği projelerle ve politikalarla artırılmaya çalışılmaktadır. Birçok çalışmada açık kaynak kodlu yazılım konusunda özel sektör için de devletlerin teşvik politikalarının bulunduğu görülmektedir. (Akyıldız, 2012)

Açık kaynak kodlu yazılımların avantajları arasında indirgenmiş yazılım masrafları, yazılım kodlarının şeffaf olmasından ötürü artan güvenlik ve azalan yazılım korsanlığı sayılabilmektedir. Açık kaynak yazılım iş birliği esas alınarak geliştirilir ve gelir kaygısı olmadan yürütülen projeler daha sürdürülebilir olur. Bu yapılanma ortak hedef güden grupların kaynak kod aracılığıyla beraber çalışma imkânı sunar.

Kamu kurumlarında açık kaynak kodlu yazılımların tercih edilmesine bağlı olarak yazılımsal maliyetlerden büyük bir tasarruf sağlanmaktadır. Geleneksel yazılımlara kıyasla işletim sistemlerinin ve ofis yazılımlarının daha kaliteli ve ucuz olması, kullanıcı bilgisayarlarının ve sunucuların daha uzun ömürlü olmaları, lisans maliyetlerinin düşük olması sayesinde uzun vadede maliyetlerde düşüş sağlanmaktadır. (Çavuş & Kurt, 2017)

3.4. Yazılımın Kurulumu

BigBlueButton yazılımı beraberinde çok sayıda bileşen ile gelir. Her bir bileşenin tek tek kurularak ayarlamalarının yapılması hem çok güçtür hem de hata olasılığını çok yükseltmektedir. Bu problemlere karşılık geliştirme ekibinin oluşturduğu bbb-

install.sh betiđi bu problemleri büyük ölçüde ortadan kaldırmaktadır. Betik, tek komutla tüm bu yazılım bileşenlerini tek tek kurarak BigBlueButton yazılımını temel düzeyde çalışır hale getirir. İlgili betik kurulumu tamamladıktan sonra güvenli bağlantının kurulabilmesi için gerekli sertifikaları ücretsiz bir servisten elde eder. Bağlantı problemlerinin ortadan kaldırılması için sunucunun önünde bulunan güvenlik duvarında TCP ve UDP portlarında gerekli izinlerin verilmesi gerekmektedir. Yazılımın ilk kurulum komutu Şekil 3.3.'te verilmiştir.

```
wget -qO- https://ubuntu.bigbluebutton.org/bbb-install.sh | bash -s -- -
v bionic-240 -s vc30.uzep.org -d
```

Şekil 3.3. BigBlueButton kurulum tetikleme komutu.

3.5. Yardımcı Araçların Kurulumu

En temel seviyede kurulum işlemi tek bir komutla gerçekleştirilse dahi beraberinde kullanılması gereken araçların kurulum aşamaları çok daha karmaşıktır. Birçok kurulum aşamasından oluşan bu süreç tamamlandıktan sonra herbirinin tek tek ayarlamaları yapılmalıdır.

3.5.1. Yük dengeleyici kurulumu

Yük dengeleyici yazılım, büyük ölçekli kullanımlar için vazgeçilmez durumdadır. Eğer kurumların kendine ait çözümleri bulunmuyorsa tercih edilen çözüm sıklıkla Scalelite yazılımıdır. Scalelite çalışmak için konteyner teknolojisini kullanır. Ayrı bir Linux sunucu üzerine konumlandırılan yazılım, gerekli konteyner paketlerinin kurulumu ardından çalışmaya hazır hale getirilir. Çalışır hale getirilen Scalelite yazılımı ardından BigBlueButton sunucularını tanıyabilmesi için her birinin tek tek içerisine tanımlanması gerekmektedir.

3.5.1.1. Redis veritabanı

Redis, Açık kaynak kodlu anahtar-değer mantığıyla çalışan bir veritabanı yazılımıdır. (Biroğul & Koçer, 2018) Hızlı erişilebilir olması Redis veritabanını önbellekleme amacıyla kullanılabilir kılmıştır. Scalelite yük dengeleyici yazılımı üzerinden geçen toplantı kayıtlarını onlara hızlıca geri çağırmak için Redis veritabanını kullanır. Redis veritabanı yazılımının birçok kurulum yöntemi bulunsa dahi Scalelite yazılımındaki rolü için konteyner teknolojisi aracılığıyla kurulmaya oldukça uygundur.

3.5.1.2. Postgre veritabanı

PostgreSQL, SQL standart sorgu dilini destekleyen ve veritabanı için ilişkisel bir model kullanan bir veritabanı yönetim sistemidir. Dünyanın en büyük ve en güvenilir açık kaynaklı veritabanı yazılımlarından biridir. (Sabah, 2016) Bu veritabanı yazılımının çok geniş kullanımı olmasına karşın Scalelite içerisinde toplantı kayıtlarının bilgilerini tutmada kullanılır.

3.5.2. TURN ve STUN sunucu kurulumu

TURN ve STUN sunucular BigBlueButton yazılımının bir güvenlik duvarı arkasında konumlandırıldığında gerekli olmaktadır. Geniş ölçekli kullanımda farklı tip ağlardan gelen bağlantıları sağlıklı olarak sağlayabilmek için kullanımı şarttır. TURN ve STUN sunucular Linux işletim sistemlerindeki paket yöneticilerinden hızlıca kurulabilir. Bu sunucuların konfigrasyonu için bir miktar bilgisayar ağları bilgisi gerekebilir ve bu konfigrasyonlar iki taraf için de ayrı ayrı yapılmalıdır.

3.6. İnce Ayarlamaların Yapılması

İlgili yazılım canlı ortamda kullanılmaya başlamadan önce bir takım ince ayar ve özelleştirmeye ihtiyaç duyar. Bunlar kimi zaman teknik değişiklikler olabilirken kimi zaman ise kişiselleştirmeye yönelik değişiklikler olabilmektedir. BigBlueButton yazılımı bu açıdan çok geniş alanda özelleştirme ve teknik değişikliğe olanak sağlar.

Canlı ortamda kullanılan sistemler için yapılması gereken deęişiklikler genellikle ařaęıdaki gibidir;

- Aę ayarları dzenlemeleri.
- Kamera ve mikrofon yetkileri.
- Kullanıcıların zaman ařımı sreleri.
- Karřılama mesajı.
- Varsayılan arkaplan.
- Video kodeęi.
- Kamera kalitesi.

Bu deęişiklikler hem platformun optimize kullanılması hem de istenilen zelleřtirmelerin yapılması iin gereklidir. Her bir deęişiklik maddesi ayrı ayrı dosyalar iinde parametrelerin deęiřtirilmesiyle yapılmaktadır. Bu deęişikliklerin yapılması kimi zaman en az kurulum sreci veya kurulum srecinden daha uzun srebilmektedir. Tek sunucu iin yapılan onlarca deęişiklik problem olmazken sunucu sayısı arttıęında bir otomasyon ihtiyacı zorunluluk haline gelmektedir.

3.7. Harcanan Efor

Kurulum ařamaları otomatikleřtirilen uzaktan eęitim altyapısında temel ama elde edilecek zaman kazanımıdır. İnsan eli ile manuel yapılan iřlemler bir defaya mahsus otomatik hale getirildięinde kurulumu her ihtiya duyulması sırasında ciddi bir zaman tasarrufu saęlamaktadır. Tekrar eden her iřlemin otomatikleřtirilmesi, o iřleme her ihtiya duyulması oranında daha fazla zaman kazandırmaktadır. Ortalama Linux bilgisine sahip bir sistem yneticisinin kurulum iřleminin her ařamasında harcadıęı yaklařık efor řu řekildedir.

- Sanal sunucuların oluřturulması ve iřletim sisteminin kurulması. (30 dakika ve tekrar eden her iřlem iin 10 dakika.)
- Gvenlik duvarı ayarlarının yapılandırılması. (15 dakika.)
- BigBlueButton kurulumunun yapılması. (Her bir sunucu iin 30 dakika.)

- Yük dengeleyici kurulması. (30 dakika ve her sunucunun tanımlanması 10 dakika.)
- Ortak depolama alanı oluşturulması. (30 dakika ve her sunucuya tanımlanması 10 dakika.)
- Turn sunucu kurulması. (30 dakika ve her sunucuya tanımlanması 10 dakika.)
- Alan adları temin edilmesi. (Her sunucu için 1 dakika.)
- Özelleştirmeler yapılması. (Her bir sunucu için 30 dakika.)

İdeal bir ortamın ihtiyaç duyduğu 10 adet BigBlueButton sunucusu, 1 adet yük dengeleyici sunucu, 1 adet TURN sunucu, 1 adet NFS paylaşımının bulunduğu yapının yaklaşık oluşturulma süresi manuel yöntemlerle 1000 dakika üzerinde gerçekleşmektedir. Belirtilen rakamlar Tubitak 120K200 numaralı "Yükseköğretimde Yeni Bir Uzaktan Öğretim Modeli ve Uygulama Platformu" başlıklı projede deneyimlenerek elde edilmiştir.

BÖLÜM 4. ÖNERİLEN YÖNTEM

Bilgisayar sistemleri yönetirken ağırlıklı olarak görsel arayüz kullanılır. Görsel arayüz esasında önceden belirlenmiş komutların bir arayüze giydirilmiş halidir. Sık kullanılan özellikler dışında görsel arayüzden düzenlenemeyecek karmaşıklıkta ayarlara müdahalede veya tekrar eden işlemlerde yine komut satırına ihtiyaç duyulabilir. Windows sistemler görsel arayüz açısından açık kaynak kodlu Linux sistemlere göre daha gelişmiştir. Arayüz esnekliğinin az olması nedeniyle Linux sistemlerde komut satırı ve otomasyon betikleri çok daha sık kullanılmaktadır. Otomasyon betikleri ve konfigrasyon araçlarına tek sunucu yönetilirken sık başvurulmazken yönetilen sunucu sayısı arttığında kullanılması kaçınılmazdır. Defalarca tekrar eden işlemlerde bir kereye mahsus hazırlanan betik gelecek zamanlarda vakitten zaman kazandırmaktadır. Zaman kazandırmasının yanında bu otomasyon betikleri aynı zamanda dokümantasyon niteliğinde de kullanılmaktadır. Betik dili okumasını bilen herkes bu betikleri inceleyerek kurulum aşamaları hakkında kolaylıkla bilgi edinebilir. Kurulum aşamalarını otomatize eden bu betikler araştırmanın asıl konusudur.

4.1. Kullanılan Araçlar

Bir ortamda çalışan yazılım farklı ortamlarda çalışmayabilir. Farklı ortamların test edilme sürecinde o ortamın hazırlanması, gerekli programların kurulması zahmetli bir süreçtir. Bir diğer bakış açısıyla yazılım ortamlarının kaynak ihtiyaçlarına göre hızlıca ölçeklenebilmesi bu otomasyon işlemine bağlıdır. Kurulu sistemlerin her birinin bakım ve güncellemelerinin tek bir noktadan otomatik olarak yapılabilmesi yine kod olarak altyapı konusu ile ilgilidir. Kod olarak altyapı yöntemi araçların yeteneğine bağlı olarak birçok sağlayıcı ile çoğu işletim sistemi aracılığıyla neredeyse bilinen tüm sistem donanımlarını yönetebilir.

4.1.1. Ansible

Ansible çok yaygın kullanılan altyapı otomasyon araçların bir tanesidir. Red Hat tarafından desteklenir. Ansible uzaktaki makinelerin hepsine tek seferde istenilen paketlerin kurulabilmesi, paket güncellemeleri, gerekli komutların çalıştırılması gibi zahmetli işlemleri yapar. Yapının büyüklüğüne göre sistem yöneticisinin elinde birçok çalışır sistem bulunabilir. Bu sistemlerde bahsedilen işlemleri elle yapmak yerine bunu bir defaya mahsus betik haline getirerek tek emirle hepsinde işlenebilir. Ansible gücünü Python'dan alır. Bağlanacağı makinelerde SSH aktif hale getirilmiş ve Python kurulmuş olması gerekmektedir. Uzak makinelere ajan kurma ihtiyacı bulunmaması Ansible aracını güvenilir kılar.

4.1.2. Terraform

Terraform altyapıların hızlı ve otomatikleştirilmiş şekilde oluşturulmasını, genişletilmesini, ölçeklenebilmesini ve versiyonlanmasını sağlar. Bir seferlik oluşturulacak betik kodu her seferinde yapılması gereken kurulum, ayarlama, güncelleme gibi zahmetlerden kurtarır. Terraform aracı popüler ve popüler olmayan, bulut ve lokal neredeyse tüm sağlayıcılarla çalışabilmesi onu güçlü kılmıştır. Terraform sayesinde bir sistemin kurulumu, kaynakların belirlenmesi, imajların seçilmesi, kurulumdan sonra gerekli komutların çalıştırılması, o sistemin önüne bir yük dengeleyici eklenebilmesi veya ihtiyaçlara göre eklenen sistemin dahil olduğu güvenlik duvarında ayarlamaların yapılabilmesi gibi olanaklara imkân tanır.

4.1.3. BASH Kabuğu

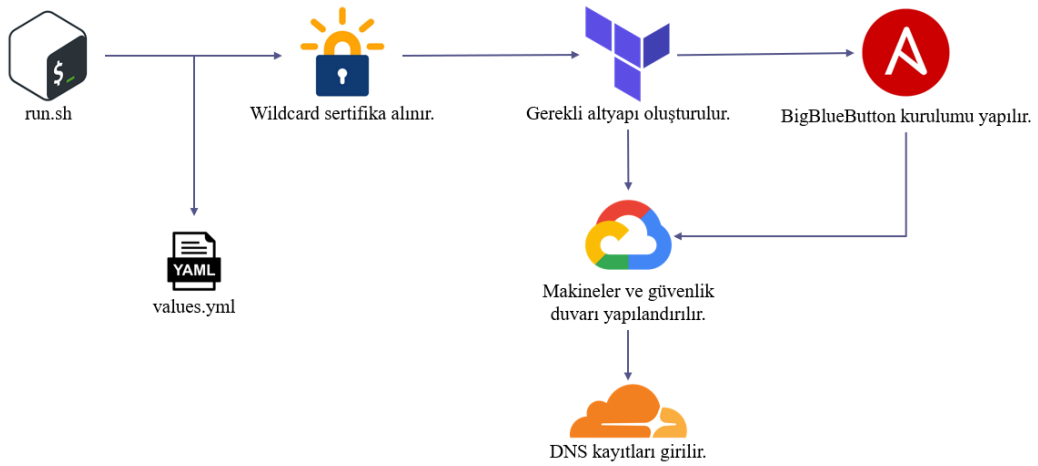
Kabuk, işletim sistemi ile kullanıcı arasında bir arayüz programıdır. Komut yorumlayıcı olarak kullanılır. Shell, kullanıcının girdiği satırı bir komut olarak okur ve kontrol edip yorumladıktan sonra programı yeni bir süreç olarak çalıştırmaya başlar. UNIX sistemlerinde kullanılan ilk Shell programları SH ve CSH Shell programlarıdır. Daha sonra KSH ve TCSH kabukları geliştirildi. BASH ilk olarak 1989 yılında GNU

projesi için geliştirildi ve Linux dağıtımları için varsayılan kabuk olarak kullanılmaktadır. (Göktaş & Göktaş, 2011)

Her programlama dilinde olduğu gibi BASH kabuğunda da döngüler ve koşullar bulunmaktadır. Birçok Linux terminalinde varsayılan olduğu için Linux sistemler ile içi içe olan kişiler BASH betik yazmakta zorlanmamaktadır. BASH betiklere sistem ve altyapılarının otomatikleştirilmesinde sıkça başvurulur.

4.2. Kurulum

BigBlueButton kurulumu tek başına oldukça kolay bir kurulumla sahip olsa da sahip olduğu bağımlılıklar nedeniyle sağlıklı kullanım için gerekli tüm aşamaların tamamlanması bir hayli karışık olmaktadır. Tüm kurulum aşamalarının otomatikleştirilmesi operasyonların hızlıca gerçekleştirilmesi açısından oldukça önemlidir. Oluşturulan yapıda kullanılan BASH betiği öncelikli olarak değerleri okuyarak sırayla kurulum araçlarını istenilen nitelikte tetikler. Let's Encrypt servisinden SSL sertifikaları elde ettikten sonra Terraform aracı tetiklenir ve makineler oluşturulur. Son aşamada ise Ansible aracı yardımıyla oluşturulan makinelerin içerisine BigBlueButton kurulumu yapılır. Otomatikleştirilen aşamalar arasında gerekli altyapıların oluşturulması, BigBlueButton kurulumu ve gerekli bağımlılıkların oluşturulmasıdır. Kurulum sürecinin adımları Şekil 4.1.'de gösterilmiştir.



Şekil 4.1. Kurulum süreci.

4.2.1. Gereksinimlerin belirlenmesi

Kuruluma başlamadan önce orta ve büyük ölçekli hizmet verebilmek için gereksinimlerin eksiksiz olması gerekmektedir. Her ne kadar tek bir BigBlueButton sunucusu tüm yeteneklerini tek bir sunucu üzerinde gerçekleştirebiliyor olsa dahi ölçek büyüdükçe yapıyı dağıtık hale getirmek yönetimini kolaylaştırmaktadır. Gereksinimlerin belirlenmesinde en büyük kriter hizmet verilecek toplam öğrenci sayısıdır. Toplam sayı doğrudan işlem gücünü etkilemesinden ötürü yatay olarak sunucu sayısını ve dolayısıyla da maliyeti etkileyecektir. BigBlueButton yazılımı geliştiricileri tarafından önerilen minimum işlem gücü ve bellek miktarı 8 vCPU ve 16 GB şeklindedir. Bu miktarda kaynak yaklaşık 150 katılımcıya eşzamanlı olarak kararlı hizmet verebilmektedir. Talep edilen katılımcı sayısı arttırılmak istendiğinde yapının sürdürülebilir olması için dikey bir genişleme yerine eş özelliklerde aynı sunucunun sağlandığı yatay genişleme önerilmektedir. Çok sayıda toplantı sunucusunun bulunacağı yapıda kayıtların ortak bir alanda saklanması, bu kayıtların tek bir noktadan yönetilmesi ve güvenilirliği için gereklidir. Toplantı içerisinde görüntü, ekran paylaşımı gibi kullanılan araçların çeşitliliği kayıtların boyutunu oldukça esnetebilmekte ancak bir ders kaydının 90 dakika olduğu varsayılarak küçük ve orta ölçekli kurumlar için dönem başına minimum 4 TB disk alanı gereksinimi yeterli görülmektedir. Bunların dışında ölçeklenebilir ve yatay olarak planlanan bir kümede işlem yükünü adil olarak dağıtmak için bir yük dengeleyici kullanmak şarttır. Yük dengeleyici yapı yüksek işlem gücü gerektirmediği için temel 2 vCPU ve 4 GB bellek yeterli olmaktadır. Son olarak katılımcı kitlesi belirlenerek bir TURN sunucu kullanılması isteğe bağlıdır. Yazılımın ihtiyaç duyduğu gereksinimler Tablo 4.1.'de verilmiştir.

Tablo 4.1. BigBlueButton donanım gereksinimleri.

CPU	MEMORY	DISK	OS
8 vCPU	16 GB	500 GB	UBUNTU 18.04

4.2.2. Farklı ortamlar için betiklerin hazırlanması

Bir uzaktan eğitim altyapısının belirlenmesinde gereksinimler kadar ihtiyaçlar da önemlidir. Hizmet verilecek öğrenci sayısı doğrudan işlem gücüne etki etmektedir. Bu noktada hizmetin kararlı bir şekilde ilerlemesi için kaynakların doğru ayarlanması ile bağlantılıdır. Yukarıda tartışılan bilgilerden faydalanılarak bir BigBlueButton sunucusunun eşzamanlı olarak yaklaşık 150 öğrenciye hizmet verebildiği göz önünde bulundurularak yaklaşık donanım ihtiyacı konusunda tahmin yapılabilmektedir. Bunun yanında büyük yapılarda çok işlem gören veritabanı gibi uygulamaları ayrık olarak barındırmak kararlılığın artmasında yardımcı olacaktır. Ölçeklerine göre özet gereksinimler aşağıdaki gibidir;

1. Küçük ölçekte kurum ve işletmeler. (150 eşzamanlı katılımcı)
 - Bir adet BigBlueButton sunucusu.
2. Orta ölçekte kurum ve işletmeler. (750 eşzamanlı katılımcı)
 - Beş adet BigBlueButton sunucusu.
 - Yük dengeleyici.
 - 4 TB NFS paylaşımı.
3. Büyük ölçekte kurum ve işletmeler. (3000 eşzamanlı katılımcı)
 - Yirmi adet BigBlueButton sunucusu.
 - Yük dengeleyici.
 - 20 TB NFS paylaşımı.
 - TURN sunucu.
 - PostgreSQL ve Redis sunucu.

4.2.3. BASH betik aşaması

Oluşturulan BASH betiği tüm işlemlerin beyni olarak düşünülmüştür. Sırasıyla aşamaları çalıştırır ve olağandışı bir durum olduğunda işlemi durdurur. Betik öncelikle yapılandırmayı okur ve bu yapılandırmaya göre sertifikaları elde eder. Sertifikalar kaydedildikten sonra Terraform aracı çalıştırılarak gerekli ağ ve altyapı ortamı uzak

bulut sunucuda oluşturulması sağlanır. Gerekli ortam sağlandıktan sonra kurulumların yapılması için işlem Ansible aracına devredilir ve kurulum tamamlanır.

4.2.4. Terraform aşaması

Terraform aracı öncelikle uzak bulut sağlayıcıya erişimi kontrol ettikten sonra eğer problem yoksa sırasıyla bir sanal ağ oluşturur ve güvenlik duvarı ayarlarını gerektiği gibi yerine getirir. Ağ yapısı hazırlandıktan sonra belirlenen sayıda BigBlueButton, Scalelite ve TURN sunucuları için gerekli makineleri oluşturur. Gerekli makineler oluşturduktan sonra sektörde bilindik bir servis aracılığıyla oluşturulan makinelerin DNS kayıtlarını yapar. Aracın yaptığı işlemler kısaca;

- Güvenlik duvarı oluşturulması ve ayarlamalarının yapılması.
- Belirlenen sayıda BigBlueButton sunucularının yanında Scalelite ve TURN sunucuları için gerekli makinelerin oluşturulması.
- Oluşturulan makinelere ait DNS kayıtlarının yapılması.

4.2.5. Ansible aşaması

Ansible aracı oluşturulan makinelere giderek BigBlueButton, Scalelite ve TURN kurulumlarını gerçekleştirir. Kayıtların tutulması için gerekli NFS kurulumunu yapar ve son aşamada ise kurum için gerekli özelleştirmeleri gerçekleştirir. Bu özelleştirmeler arasında gösterilecek varsayılan mesajlar, varsayılan sunumlar, ağ ayarları sayılabilir. Tüm ayrıntılarıyla kurulum aşamaları aşağıdaki gibidir;

1. Sertifikaların yüklenmesi.
2. BigBlueButton kurulumu.
 - Bağımlılık paketlerinin kurulması.
 - Dil ve bölge ayarlarının yapılması.
 - Güvenlik duvarı ayarlarının yapılması.
 - BigBlueButton yazılımının kurulması
3. Scalelite yük dengeleyici kurulumu

- Konteyner aracının kurulması.
 - Konteyner yardımcı aracının kurulması.
 - Scalelite yazılımının konteyner aracılığıyla ayağa kaldırılması.
 - BigBlueButton sunucularının Scalelite içerisine kaydedilmesi.
4. TURN sunucu kurulumu.
- TURN yazılımının kurulması.
 - TURN konfigrasyonlarının yapılması.
 - BigBlueButton sunucularının TURN parametlerinin ayarlanması.
5. Takip ajanlarının kurulumu.
6. Özelleştirmelerin yapılması.
- Yardımcı betiklerin sunuculara yüklenmesi.
 - Varsayılan dokümanın değiştirilmesi.
 - Ağ bağlantısı ayarlarının iyileştirilmesi.
 - Karşılama mesajının değiştirilmesi.
 - Otomatik çalışacak görevlerin tanımlanması.
 - Gereksiz paketlerin kaldırılması.

4.3. Yazılım Bakımı

Çok sayıda kullanıcıya hizmet veren büyük ölçekteki bir çevrimiçi toplantı yazılımında düzenli olarak gerçekleştirilmesi gereken birtakım işlemler vardır. Yazılım kararlı ve sürekli çalışmaya devam edebilmesi için bu bakım işlemleri mecburidir. Düzenli olarak bakımı yapılmayan yazılım kararlılığını kaybedebilir, hizmet vermeyi durdurabilir veya güvenlik zafiyetlerine hedef haline gelebilmektedir. Belirli aralıklarla ve düzenli kontrollerle yapılan bakım işlemleri yazılımın sürekliliğini garanti altına alacaktır.

4.3.1. Sertifikaların yenilenmesi

Uçtan uca güvenliği sağlamak için SSL şifreleme kullanan BigBlueButton yazılımı HTTPS bağlantı kurmak için sertifikalara ihtiyaç duyar. Bu sertifikalar ücretli olarak temin edilebildiği gibi Let's Encrypt gibi araçlar vasıtasıyla ücretsiz olarak da temin

edilebilir. Sınırlı süre geçerli olan bu sertifikalar hizmet vermeye devam edebilmesi için zamanı dolmadan yenilenmelidir. Ücretli sertifikalar nispeten uzun ömüre sahip olsa da Let's Encrypt tarafından temin edilen sertifikalar 3 ay ömre sahiptir ve dolayısıyla da sıklıkla yenilenmesi gerekir. Yapılan çalışmada oluşturulan modüler kurulum betikleri daha sonra yeniden kullanılabilir. Sertifikaların yüklenmesi aşaması tekrar çalıştırılarak sertifikaların yenilenmesi kolaylıkla ve toplu bir şekilde yapılabilir.

4.3.2. Biriken dosyaların temizlenmesi

Belirli süre çalışan her yazılım gibi çevrimiçi toplantı yazılımları da çok sayıda veri üretir ve düzenli aralıklarla geriye dönük temizlenmeye ihtiyaç duyar. Bunların arasında başlıca günlük dosyaları ve ham toplantı kayıt dosyaları bulunmaktadır. Yazılım çalıştığı sürece meydana gelen olaylar, hatalar, ders dökümleri gibi baştan sonra tüm bilgilerin tutulduğu günlük dosyaları disk alanını işgal etmemesi için düzenli olarak temizlenmelidir. Bir diğer düzenli olarak temizlenmesi gereken dosyalar ise ham kayıt dosyalarıdır. Kayıtların kullanıcıya ulaştırılmadan önceki ham halleri belirli süre saklanmaktadır. Ham kayıt verileri hatalı dönüştürme işlemi, ana kayıtların kayıba uğraması, kayıtların metodlarla dönüştürülmek istenmesi gibi nedenlerden ötürü bilinçli olarak tutulur. Linux sisteminin bir fonksiyonu olan görev zamanlayıcısı ile kontrol edilen bu temizlik işleminde, hangi günden eski verileri tutacağı betikler tarafından yürütülür. Görev zamanlayıcıya ait dosya yolu Şekil 4.2.'de verilmiştir. Sunucuların aktivitesine göre çok daha hızlı yer kaplamaya başlayan günlük ve ham kayıt dosyaları varsayılan olarak 15 gün tutulur ancak ihtiyaca göre bu rakam kısaltılabilir veya uzatılabilir.

```
/etc/cron.daily/bigbluebutton
```

Şekil 4.2. Temizlik yapan betiğin görev zamanlayıcı içerisindeki dosya yolu.

4.3.3. Paket gncellemeleri

Dzenli olarak gncellenen yazılımlar hem optimizasyon aısından iyileřtirmeler barındırır hem gvenlik aıklarının kapatılmasını saęlar hem de eřitli fonksiyonlar bnyesine katabilir. BigBlueButton yazılımı aık kaynak kodlu bir yazılımdır ve topluluk tarafından dzenli olarak gncellenir. Canlı ortamda gncellemelerin yapılması riskli bir iřlemdir ve iřlem gerekleřtirilirken bazı nlemlerin alınması gerekmektedir. Yapılan alıřmadaki kullanılan ortamda yazılım gncellemeleri Ubuntu iřletim sisteminin paket yneticisi tarafından yapılıp srm olarak gncel hatalardan kaınmak amacıyla bir srm geriden takip edilmektedir. ncelikle tek sunucuda yapılan gncelleme, sunucu gzlenerek sorun ıkarmaması durumunda tm sunuculara yayılmaktadır. Gncelleme iřlemi yapılan bazı zelleřtirmeleri etkileyebildięi iin sık deęiřen zelleřtirmeleri yeniden uygulayabilmek iin betikler hazırlanmıřtır. Her gncelleme iřlemi sonrasında bu betik alıřtırılarak yapılan zelleřtirmeler yeniden uygulanmaktadır.

4.4. Zaman Kazanımı

Bir evrimii uzaktan eęitim sisteminin kod olarak altyapı disipliniyle otomatik olarak oluřturulması sreci kolaylařtırmasının yanında byk zaman tasarrufu saęlamaktadır. Yapıyı oluřturmak isteyen kullanıcı onlarca adım yerine sadece betięi indirerek gerekli deęiřkenleri ayarladıktan sonra kurulumu bařlayabilmektedir. Kurulum sreci alıřma boyunca yapılan lmlerde paralel iřlemlerin de yardımıyla yaklaşık olarak 1 saatte bitmektedir. Onlarca adım ve harcanan efor ile birlikte klasik yntemle 1000 dakikalara bulan kurulum sreci 60 dakikalara indirilmiřtir. Yapılan alıřma Tubitak 120K200 numaralı "Yksekęretimde Yeni Bir Uzaktan ęretim Modeli ve Uygulama Platformu" bařlıklı projede uygulanarak sistemin olduka hızlı bir Őekilde ayaęa kalkmasına yardımcı olmuřtur.

BÖLÜM 5. SONUÇLAR

İçerisinde bulunduğunuz pandemi koşulları tüm yaşam şartlarını değiştirdiği gibi eğitim olanaklarını da değiştirmiştir. Bu şartlar altında eğitim büyük miktarda uzaktan yöntemlere geçmek zorunda kalmış ve birçok kurum uzaktan eğitim ortamlarına ihtiyaç duymuştur. Kısa zamandaki bu geçiş ve değişken ihtiyaçlar bu ortamları yönetmeyi oldukça güçleştirmiştir. Eğitim kurumları hazır çözümleri kiralamayı tercih edebilirken diğer yandan açık kaynak yazılımları kendi bünyesine katmayı denemektedirler. Bu yazılımların kurum içerisine entegre edilmesinde değişken ihtiyaçlara kısa sürede karşılık verebilmek o birimin yöneticileri için oldukça güçtür. Bu çalışmanın amacı olan uzaktan eğitim ortamlarının tamamıyla otomatik olarak kurulması hedefi, birim sorumlularının oldukça vaktini alan ve sürekli tekrar eden işlemleri hızlandırarak taleplere çevik hizmet edebilmektir.

Yaşam şartlarında meydana gelen zorunlu değişim eğitim olanaklarını da baştan aşağı etkilemiştir. Pandemi koşullarında yüzyüze eğitim göremeyen öğrenciler sayılı bir takım çevrimiçi toplantı yazılımına bağlı kalarak süreçlerini ilerletmişlerdir. Eğitim özelinde tasarlanmış toplantı yazılımlarının eksikliği öğrencilerin eğitim kalitesini maalesef olumsuz etkilemiştir. Bu eksiklik, bir grup açık kaynak yazılım geliştiricisi tarafından farkedilerek ortaya BigBlueButton yazılımı çıkarılmıştır. Açık kaynak kodlu, lisans maliyetleri olmayan, eğitim özelinde geliştirilmiş, hızlı ve kullanımı kolay bir yazılım olan BigBlueButton her ne kadar bu özellikleri barındırsa da ilk kurulum ve düzenli olarak yönetim hizmetine muhtaçtır. Hizmet verilen topluluk ve yönetilen sunucu sayısı az olduğunda kolaylıkla kontrol edilebilirken sayılar arttıkça kontrol imkânı zorlaşmaktadır. Bu ihtiyaca istinaden geliştirilen otomasyon betikleri gerek yazılımın kurulum aşamasında gerek ise yönetilmesinde büyük zaman tasarrufu sağlamaktadır. BigBlueButton ortamının çok kısa sürede ayağa kaldırılmasının yanında oluşabilecek felaket senaryolarında hızlı bir şekilde çalışan en kararlı haline

geri döndürülmesine yardımcı olmaktadır. Yapılan çalışma gerçek bir kurumda yürütülmüş olup çok sayıda öğrenciye hizmet verilmesine yardımcı olmuş ve karşılaşılan problemlere oldukça hızlı bir şekilde reaksiyon alınmasını sağlamıştır. Sistemin sürekliliğini sağlamak için sunucu kaynaklarından toplanan veriler görselleştirilmiş ve belirlenen kriterler çerçevesinde alarmlara bağlanmıştır. Belirlenen seviyelerin üzerine çıkan parametreler önceden bildirilerek sistemin sürekliliği korunmuştur. Toplantı kayıtlarından elde edilen aktivite verileri daha sonra işlenmek üzere uygun formatta saklanmıştır.

Tablo 5.1. Klasik kurulum ve kod olarak altyapı kurulum farkı.

Klasik Kurulum		Kod Olarak Altyapı Kurulum	
Sanal sunucular oluşturulur ve işletim sistemi kurulur. (30 dakika ve tekrar eden her işlem için 10 dakika.)	14 işlem	Gerekli değişkenler ayarlanır. (15 dakika.)	1 işlem
Güvenlik duvarı ayarları yapılır. (15 dakika.)	1 işlem		
BigBlueButton kurulumu yapılır. (Her bir sunucu için 30 dakika.)	10 işlem		
Yük dengeleyici kurulur. (30 dakika ve her sunucunun tanımlanması 10 dakika.)	1 + 10 işlem		
Ortak depolama alanı oluşturulur. (30 dakika ve her sunucuya tanımlanması 10 dakika.)	1 + 10 işlem	Betik başlatılır ve kurulum tetiklenir. (45 dakika.)	1 işlem
Turn sunucu kurulur. (30 dakika ve her sunucuya tanımlanması 10 dakika.)	1 + 10 işlem		
Alan adları temin edilir. (Her sunucu için 1 dakika.)	14 işlem		
Özelleştirmeler yapılır. (Her bir sunucu için 30 dakika.)	10 işlem		
Toplam 1179 dakika. (2.5 mesai günü.)		Toplam 60 dakika.	

Uzaktan eğitim ortamını oluşturmak, tüm bileşenleriyle birlikte çok sayıda aşamadan oluşan bir süreçtir. Her adımı farklı birimleri ilgilendiren bu süreç, ortama ihtiyaç duyulduğunda en kısa ve kolay şekilde sürdürülmelidir. Donanım kaynakları ve açık kaynak kodlu yazılımların oluşturulmasını çeşitli araç ve betikler ile nasıl otomatikleştirileceğini araştıran bu çalışma Sakarya Üniversitesi bünyesinde aktif

olarak kullanılmaya başlanmıştır. Tek başına manuel adımlarla yaklaşık 1000 dakikaları bulan ortamın kurulum işlemi, bu adımların otomatikleştirilerek sürenin 60 dakikalara inmesi sağlanmıştır. Dahası bu işlem paralel çalıştırılabildiği için geniş kapsamlı kurulumlarda vakit kazancının daha çok artması sağlanmıştır. Konfigrasyon değişiklikleri sırasında dosya yapılandırmalarından elle düzenlenen veriler de otomatik betiklerle desteklendiğinde çok sayıda sunucuda kaybedilen zaman kaybı engellenerek sistem yöneticisinin tekrar eden işlemlerle vakit kaybetmesi önlenmiştir.

Çalışmanın öncelikli amaçlarından birisi olan kurulum aşamalarında zaman kazanımı konusunda kayda değer iyileştirme elde edilmiştir. Manuel yöntemler ile kurulan ortam için hem çok ciddi efor gerektirmesi hem de aşamaların fazla olmasından dolayı insan hatasına müsait olması çalışmanın katkılarını ortaya koymaktadır. Kod olarak altyapı disiplini ile oluşturulan yapı klasik kurulumla göre çok daha az aşamadan oluşmaktadır. İdeal ortamın ihtiyaç duyduğu kaynak miktarınca iki kurulum yöntemi arasındaki aşamaların farkı Tablo 5.1.'de gösterilmiştir. Kurulum aşamalarının çeşitli araçlar yardımıyla otomatikleştirilmesi gerek kurulum aşamalarında gerek ise ortamın yeniden oluşturulmasında sistemin sürekliliğini kazanması konusunda oldukça zaman kazandırmaktadır.

KAYNAKLAR

- Akpınar, Y. (2003). Öğretmenlerin Yeni Bilgi Teknolojileri Kullanımında Yükseköğretimin Etkisi: İstanbul Okulları Örneği. *The Turkish Online Journal of Educational Technology*, 79-96.
- Akyıldız, F. (2012). Kamu Yönetiminde Açık Kaynak Kodlu Yazılımlar. *C. Ü. İktisadi ve İdari Bilimler Dergisi*, 17-41.
- Akyıldız, F. (2012). Kamu Yönetiminde Açık Kaynak Kodlu Yazılımlar. *İktisadi ve İdari Bilimler Dergisi*, 17-41.
- Artsın, M., & Günal, K. (2021). Öğrenme yönetim sistemi seçiminde kullanılacak kriterlerin belirlenmesine yönelik çok ölçütlü karar verme yöntemi önerisi. *Açıköğretim Uygulamaları ve Araştırmaları Dergisi*, 87-108.
- Ayaz, S. (2019). *Json Sözdizimli Gerçek Zamanlı İletişim Protokolü*. İstanbul: Okan Üniversitesi.
- Aydın, C. Ç., & Biroğul, S. (2008). E- Öğrenmede Açık Kaynak Kodlu Öğretim Yönetim Sistemleri ve Moodle. *Bilişim Teknolojileri Dergisi*, 31-36.
- Aydın, C. Ç., & Biroğul, S. (2008). E- Öğrenmede Açık Kaynak Kodlu Öğretim Yönetim Sistemleri ve Moodle. *Bilişim Teknolojileri Dergisi*, 31-36.
- Biroğul, S., & Koçer, K. (2018). WEB ve Mobil Tabanlı Bakım Onarım ve Varlık Yönetim Sisteminde Önbellekleme Yaklaşımları. *Matematik Bilimleri ve Tasarım Dergisi*, 579-589.
- Çavuş, M. F., & Kurt, H. S. (2017). Kamu Kurumlarında Açık Kaynak Kodlu Yazılımların Kullanımı. *Siyaset, Ekonomi ve Yönetim Araştırmaları Dergisi*, 39-49.
- Demirtel, H., & Sağiroğlu, Ş. (2011). Özgür Dağıtım Konu Olan Yazılımlar: Nedir, Ne Kadar Güvenlidir? *TÜBAV Bilim Dergisi*, 52-63.
- Durak, G., Çankaya, S., & İzmirli, S. (2020). COVID-19 Pandemi Döneminde Türkiye'deki Üniversitelerin Uzaktan Eğitim Sistemlerinin İncelenmesi. *Necatibey Eğitim Fakültesi Elektronik Fen ve Matematik Eğitimi Dergisi*, 787-809.
- Göktaş, A. H., & Göktaş, R. (2011). *Bash Scripting*. Ra-tek Bilgisayar Danışmanlık İletişim Hizmetleri Tic. Ltd. Şti.

- IBM. (2020). Network File System. IBM: <https://www.ibm.com/docs/en/aix/7.1?topic=management-network-file-system> Erişim Tarihi: 08.11.2021.
- İzmirli, S., & Akyüz, H. İ. (2017). Eş Zamanlı Sanal Sınıf Yazılımlarının İncelenmesi. Eğitimde Kuram ve Uygulama, 788-810.
- Küçükönder, N., & Kır, İ. (2016). Uzaktan Eğitim Uygulamalarında Açık Kaynak Kodlu Öğrenme Yönetim Sistemlerinin Yeniden Yapılandırılmasının İncelenmesi. Kahramanmaraş Sütçü İmam Üniversitesi Sosyal Bilimler Dergisi.
- Özkul, A. E., & Aydın, C. H. (2012). Öğrenci Adaylarının Açık ve Uzaktan Öğrenmeye Yönelik Görüşleri. Akademik Bilişim.
- Sabah, L. (2016). Coğrafi Bilgi Sistemleri Kullanılarak Açık Kaynak Kodlu Kurumsal Web Uygulamalarını Geliştirilmesi. 4. Uluslararası Coğrafya Sempozyumu. Antalya: Mehmet Akif Ersoy Üniversitesi.
- Scalelite. (2022). github.com: <https://github.com/blindsidenetworks/scalelite> Erişim Tarihi: 20.03.2022.
- Sezer, C., & Korucu, A. T. (2019). Bilişim Teknolojileri ve Yazılım Dersinde Öğrenme Yönetim Sistemi Kullanımının Öğrenci Akademik Başarısına Etkisi. Gazi Eğitim Bilimleri Dergisi, 157-176.
- Sierra-Fernández, J.-M., Espinosa-Gavira, M.-J., Florencias-Oliveros, O., Espinosa-Gavira, M.-J., Gonzales-de-la-Rosa, J.-J., Agüera-Pérez, A., & Palomares-Salas, J.-C. (2021). Online System for Power Quality Operational Data Management in Frequency Monitoring Using Python and Grafana. Energies.
- Süral, İ. (2011). Çevrimiçi Toplantı ve Ders Sunum Ortamı Olarak BigBlueButton. Akademik Bilişim Konferansı, (s. 457-460). Malatya.
- Süral, İ. (2015). Açık ve uzaktan öğrenmede teknolojik altyapının oluşturulması. Açıköğretim Uygulamaları ve Araştırmaları Dergisi, 81-95.
- Tutsun, E. (2020). Öğrenme Analitikleri ve Yükseköğretimdeki Uygulama Alanları. Bilişim Teknolojileri Dergisi, 243-254.
- Uygun, D., Yıldız, S., & Hertsch, M. F. (2010). Variations and additions and for the E-Learning Source Moodle with a Perspective towards the Turkish education system. International Conference on new Horizons in Education (s. 927-932). Cyprus: INTE.
- Yıldırım, İ. S., Göktaş, Y., Temur, N., & Kocaman, A. (2004). İyi Bir Öğrenme Yönetimi Sistemi İçin Kriter Önerisi. Türk Eğitim Bilimleri Dergisi, 455-462.

EKLER

EK A: BigBlueButton ansible kurulum kodu

```
- set_fact:
  desired_hostname: "{{ inventory_hostname }}"
- name: Install Prerequisites
  apt:
    update_cache: yes
    name:
      - net-tools
      - language-pack-en
      - lsb-release
      - python3
      - python3-apt
      - python-apt
      - haveged
      - dbus
      - wget
    state: present
    force_apt_get: yes
- name: Change locale
  locale_gen:
    name: tr_TR.UTF-8
    state: present
- name: Configure firewall
  when: bigbluebutton_configure_firewall|bool
  include_tasks: firewall.yml
```

```

- name: Warn user
  run_once: yes
  debug:
    msg: >
      Ready to run wget -qO- https://ubuntu.bigbluebutton.org/bbb-install.sh | bash -s --
-v {{ bbb_version }} {{ inventory_hostname }} -d
      This takes around 15 mns
- name: Launch install script
  shell: wget -qO- https://ubuntu.bigbluebutton.org/bbb-install.sh | bash -s -- -v {{
bbb_version }} -s {{ inventory_hostname }} -d
  become: yes
- name: Install dependencies
  apt:
    name:
      - apt-transport-https
      - ca-certificates
      - curl
      - gnupg2
      - software-properties-common
    state: present
- name: Running setip
  shell: "bbb-conf --setip {{ desired_hostname }}"
- name: Configure Big Blue Button
  lineinfile:
    regexp: '^{{ item.key }}='
    line: "{{ item.key }}={{ item.value }}"
    path:/usr/share/bbb-web/WEBINF/classes/bigbluebutton.properties
    state: present
  loop: "{{ bigbluebutton_config|dict2items }}"
  notify: restart bigbluebutton
- name: Modify webrtc value

```

```

    command: yq w -i /usr/local/bigbluebutton/bbb-webrtc-sfu/config/default.yml {{
item.key }} "{{ item.value }}"
    loop: "{{ bigbluebutton_webrtc|dict2items }}"
    notify: restart bigbluebutton
- name: Change default presentation
  copy:
    src: files/default.pdf
    dest: /var/www/bigbluebutton-default/default.pdf
    mode: +r
- name: Manage camera profiles
  when: bigbluebutton_camera_profiles is defined
  block:
- name: Generate yq script for camera profile
  copy:
    dest: /var/cache/yq_cameraProfiles.yml
    content: |
      - command: update
        path: public.kurento.cameraProfiles
        value:
          {{ bigbluebutton_camera_profiles|to_nice_yaml|indent(2) }}
    notify: restart bigbluebutton
- name: Set camera profiles
  command: >-
    yq w
    -i
    -s /var/cache/yq_cameraProfiles.yml
/usr/share/meteor/bundle/programs/server/assets/app/config/settings.yml

```

EK B: BigBlueButton ansible özelleştirme kodu

- name: copy scalelite postpublish

copy:

src: files/scalelite_post_publish.rb

dest: /usr/local/bigbluebutton/core/scripts/post_publish/scalelite_post_publish.rb

owner: root

mode: +x

- name: copy batch import

copy:

src: files/scalelite_batch_import.sh

dest: /root/scalelite_batch_import.sh

owner: root

mode: +x

- name: copy rebuild-non-presentation.sh

copy:

src: rebuild-non-presentation.sh

dest: /root/rebuild-non-presentation.sh

owner: root

mode: +x

- name: copy post_publish

copy:

src: post_publish.rb

dest: /usr/local/bigbluebutton/core/scripts/post_publish/post_publish.rb

owner: root

mode: +x

- name: copy video.rb

copy:

src: video.rb

dest: /usr/local/bigbluebutton/core/lib/recordandplayback/generators/video.rb

owner: root

mode: +x

- name: copy presentation.yml
 - copy:
 - src: presentation.yml
 - dest: /usr/local/bigbluebutton/core/scripts/presentation.yml
 - owner: root
- name: Deploy scalelite scalelite.yml file
 - template:
 - src: scalelite.yml.j2
 - dest: /usr/local/bigbluebutton/core/scripts/scalelite.yml
- name: Update apt-get repo and cache
 - apt: update_cache=yes
- name: Install the package bbb-webhooks
 - apt:
 - name: bbb-webhooks
 - state: present
- name: Install the package openjdk-8-jdk-headless
 - apt:
 - name: openjdk-8-jdk-headless
 - state: present
- name: fix special characters
 - become: True
 - shell:
 - cmd:'native2ascii"/usr/share/bbb-web/WEB-INF/classes/bigbluebutton.properties"
"/usr/share/bbb-web/WEB-INF/classes/bigbluebutton.properties"'
- name: Remove bbb-demo
 - apt:
 - name: bbb-demo
 - state: absent
 - purge: yes

EK C: Yükseltme sonrası konfigrasyon değişikliği BASH betiği

```
#!/bin/bash
IP=$(curl -s http://whatismyip.akamai.com)
TARGET1=/opt/freeswitch/conf/sip_profiles/external.xml
TARGET2=/usr/local/bigbluebutton/bbb-webrtc-sfu/config/default.yml
xmlstarlet edit --inplace --update '//param[@name="ext-rtp-ip"]/@value' --value
"\${external_rtp_ip}" $TARGET1
xmlstarlet edit --inplace --update '//param[@name="ext-sip-ip"]/@value' --value
"\${external_sip_ip}" $TARGET1
yq w -i $TARGET2 kurento[0].ip "$IP"
yq w -i $TARGET2 freeswitch.ip "$IP"
yq w -i $TARGET2 freeswitch.sip_ip "$IP"
yq w -i /usr/share/meteor/bundle/programs/server/assets/app/config/settings.yml
public.media.sipjsHackViaWs true
bbb-conf --restart
```

EK D: Toplantı analitikleri içeri aktaran PERL betiği

```
#!/usr/bin/ruby
require 'rubygems'
require 'fileutils'
require "trollop"
opts = Trollop::options do
  opt :meeting_id, "Meeting id to archive", :type => String
  opt :format, "Playback server", :type => String end
meeting_id = opts[:meeting_id]
system("cp $(ls -d /var/bigbluebutton/learning-dashboard/#{meeting_id}/*|head -n
1)learning_dashboard_data.json
/var/bigbluebutton/published/presentation/#{meeting_id}/learning_dashboard_data.js
on")
system("rm -rf /var/bigbluebutton/learning-dashboard/#{meeting_id}/")
```

EK E: İşlenmeyen kayıtları tekrar işleme sokan BASH betiği

```
#!/bin/bash
bbb-record --list | grep "^\.{128\} " | cut -b 1-54 >> list.txt
filename='list.txt'
while read p; do
    bbb-record --rebuild ${p}
done < $filename
```

EK F: Kayıtların dönüştürülmesinde API hizmeti veren Python kodu

```
#!/usr/bin/python
from flask import Flask, jsonify
from flask import request
from flask import make_response
from flask import redirect
from flask_cors import CORS, cross_origin
import subprocess
import os

app = Flask(__name__)
CORS(app)

@app.route('/')
def Main():
    return 'online'

@app.route('/addqueue/', methods=['GET'])
def add():
    command = 'pueue add "docker run -v /var/bigbluebutton/:/var/bigbluebutton/ -v /usr/local/bigbluebutton/core/scripts/:/usr/local/bigbluebutton/core/scripts/ bedrettinyuce/bbbrecorder-s3 -p ' + request.values.get('playback') + ' -m ' + request.values.get('externalId') + '"'
    p = subprocess.Popen(command, stdout=subprocess.PIPE, shell=True)
    out = p.communicate()[0]
    #app.logger.info(out)
    if out.find('added') != -1:
        status = 'queued'
    else:
        status = 'failed'
    params = {
```

```
'status': status,  
'playback': request.values.get('playback'),  
'externalId': request.values.get('externalId')  
}  
return jsonify(params)  
if __name__ == "__main__":  
    app.run(threaded=True, debug = True, host = "0.0.0.0")
```

EK G: Kayıtların dönüştürülmesini tetikleyen PERL kodu

```
#!/usr/bin/ruby
require 'yaml'
require 'rubygems'
require 'aws-sdk-s3'
require 'fileutils'

if !Dir.exist?('/var/bigbluebutton/record_mp4')
  FileUtils.mkdir_p '/var/bigbluebutton/record_mp4'
  FileUtils.mkdir_p '/var/bigbluebutton/record_mp4/log'
  FileUtils.mkdir_p '/var/bigbluebutton/record_mp4/temp'
  FileUtils.mkdir_p '/var/bigbluebutton/record_mp4/uploaded'
end

class FileQueue
  def initialize(file_name)
    @file_name = file_name
  end

  def pop
    value = nil
    rest = nil
    safe_open('r') do |file|
      value = file.gets
      rest = file.read
    end
    safe_open('w+') do |file|
      file.write(rest)
    end
    value
  end

  private
end
```

```

def safe_open(mode)
  File.open(@file_name, mode) do |file|
    file.flock(File::LOCK_EX)
    yield file
    file.flock(File::LOCK_UN)
  end
end

end

end

queue = FileQueue.new '/var/bigbluebutton/queue_mp4.txt'
props =
YAML::load(File.open('/usr/local/bigbluebutton/core/scripts/bigbluebutton.yml'))
creds = YAML::load(File.open('/usr/local/bigbluebutton/core/scripts/s3_creds.yml'))
host = props['playback_host']
endpoint = creds['endpoint']
access_key_id = creds['access_key_id']
secret_access_key = creds['secret_access_key']
bucket = creds['bucket']
region = creds['region']
playback = creds['playback_server']
s3Client = Aws::S3::Client.new(endpoint: endpoint, access_key_id: access_key_id,
secret_access_key: secret_access_key, region: region)
s3Resource = Aws::S3::Resource.new(client: s3Client)
bucketObj = s3Resource.bucket(bucket)

while !File.zero?("/var/bigbluebutton/queue_mp4.txt") do
  meetingId = queue.pop
  puts meetingId
  logfile = "/var/bigbluebutton/record_mp4/log/#{meetingId}.log"
  pid = spawn("docker run -v /var/bigbluebutton/record_mp4/temp:/root/records
bedrettinyuce/bbb-recorder node export.js \""" + playback +

```

```

"/playback/presentation/2.0/playback.html?meetingId=" + meetingId + "\" + "
meeting.webm 0 true", [:err] => logfile)
  Process.wait(pid)
  if(File.exist?("/var/bigbluebutton/record_mp4/temp/meeting.mp4"))
    File.rename("/var/bigbluebutton/record_mp4/temp/meeting.mp4",
"/var/bigbluebutton/record_mp4/temp/" + meetingId + ".mp4")
  end
  file_name = '/var/bigbluebutton/record_mp4/temp/' + meetingId + '.mp4'
  key = File.basename(file_name)
  puts "Uploading file #{file_name} to bucket #{bucket}..."
  s3Client.put_object(
    :bucket => bucket,
    :key   => bucket + '/' + key,
    :body  => IO.read(file_name),
    :acl   => 'public-read'
  )
  if(bucketObj.object(bucket + '/' + meetingId + '.mp4').exists?)
    File.delete('/var/bigbluebutton/record_mp4/temp/' + meetingId + '.mp4')
    FileUtils.touch('/var/bigbluebutton/record_mp4/uploaded/' + meetingId + '.done')
  end
end
end
puts "ended"

```


EK H: BigBlueButton için kullanılan zamanlanmış görevleri ekleyen ansible kodu

```
- name: clear crons
  become: True
  ignore_errors: yes
  when:
    - clear_crontabs == True
  shell:
    cmd: "crontab -r"
- name: add bbb-recorder-stop /etc/cron.d
  ansible.builtin.cron:
    name: add cron stop record service
    weekday: "1-5"
    minute: "0"
    hour: "9"
    user: root
    job: "systemctl stop bbb-rap-process-worker.service bbb-record-core.timer"
    cron_file: bbb-recorder-stop
- name: add bbb-recorder-stop-2 /etc/cron.d
  when:
    - multi_record_process == True
  ansible.builtin.cron:
    name: add cron stop record service
    weekday: "1-5"
    minute: "0"
    hour: "9"
    user: root
    job: "systemctl stop bbb-rap-process-worker-2.service"
    cron_file: bbb-recorder-stop-2
- name: add bbb-recorder-start under /etc/cron.d
  ansible.builtin.cron:
    name: add cron stop record start
```

```
weekday: "1-5"  
minute: "0"  
hour: "20"  
user: root  
job: "systemctl start bbb-record-core.timer"  
cron_file: bbb-recorder-start  
- name: add reboot-weekly /etc/cron.d  
  ansible.builtin.cron:  
    name: reboot weekly  
    weekday: "SUN"  
    minute: "30"  
    hour: "1"  
    user: root  
    job: "/sbin/shutdown -r now"  
    cron_file: reboot-weekly  
- name: copy bbb-clean cron  
  copy:  
    src: bbb-clean  
    dest: /etc/cron.daily/bbb-clean  
    owner: root  
    mode: +x  
- name: copy bigbluebutton cron  
  copy:  
    src: bigbluebutton  
    dest: /etc/cron.daily/bigbluebutton  
    owner: root  
    mode: +x
```

ÖZGEÇMİŞ

Adı Soyadı : Osman Bedrettin YÜCE

ÖĞRENİM DURUMU

Derece	Eğitim Birimi	Mezuniyet Yılı
Yüksek Lisans	Sakarya Üniversitesi / Fen Bilimleri Enstitüsü / Bilişim Sistemleri Mühendisliği	Devam Ediyor
Lisans	Sakarya Üniversitesi / Bilgisayar ve Bilişim Bilimleri Fakültesi / Bilişim Sistemleri Mühendisliği	2020
Lise	Ahmet Keleşoğlu Anadolu Lisesi	2014

İŞ DENEYİMİ

Yıl	Yer	Görev
2021-Halen	Intertech Bilgi İşlem ve Pazarlama	Sistem Mühendisi
2020-2021	Sakarya Üniversitesi	Öğrenci Asistan
2020-2020	Sakarya Üniversitesi	Proje Bursiyeri
2020-2020	Kloia Yazılım ve Danışmanlık	Dev-Ops Mühendisi

YABANCI DİL

İngilizce, Almanca